Robustness and Stability of Deep Learning

A DISSERTATION
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Chieh-Hsin Lai

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Gilad Lerman

June, 2021

Acknowledgements

Throughout my time in graduate school, I have been receiving supports and assistance from many people whom I am grateful for.

First and foremost, I would like to express my sincere gratitude to my advisor, Professor Gilad Lerman. It is Gilad that brought me in to the realm of data science. Without his invaluable advice, continuous supports and dedicated guidance, I would have never been accomplished this work. During numerous discussions with Gilad, he always raises interesting questions and feedback which challenge my thoughts and further push me to sharpen my ideas and our work. Gilad's rigorous attitude and profound insights to the research sets up an epitome for me as a great researcher.

My deepest appreciation also goes for Professor Dongmian Zou at Duke Kunshan University. He is not only my collaborator but also a mentor of my research and my life. Dongmian's supports encouraged me to overcome all the difficulties while I was struggling in the transition of the research direction from pure mathematics to data science. Dongmian is so knowledgeable and patient that he guides me to become a more mature researcher.

I would like to extend my gratitude to Professor Ju Sun from Department of Computer Science and Engineering. Professor Sun leads me into the interesting field of inverse problems via deep learning. His passion and insightful ideas always inspire me. I especially thank Prof. Sun for inviting me to join the discussion group of his team where I have been enjoying the active atmosphere of ideas exchanging and advanced research works sharing. Especially, I would like to express my truly appreciation to Kshitij Tayal, Raunak Manekar, Zhong Zhuang and Vipin Kumar. Without their efforts, our joint works on symmetries breaking of inverse problems (Chapter 4) would not have completed.

I would also like to thank Professor Wei-Kuo Chen, Ju Sun and Jeffrey Calder for serving as my committees. Their feedback on my research are invaluable.

I am grateful for my peers Yunpeng Shi, Shaohan Li and Zhengyi Gong for their important comments and discussions on my research. My special thanks also go to Yuki Mitsufuji who appreciates our works and provides insights into potential real-world applications. His encouragement pushes me to develop our works further and motivates me to think beyond the scope.

I deeply appreciate my mother, Mei-Chuan Chen. She raises me up alone with all efforts and supports me for my graduate study in the United States. Without her unconditional love and supports, I would have never had a chance to fulfill what I am pursuing for.

At last, I want to acknowledge my significant other, Ching-Yun Chang. She is always being warm, supportive, considerate and encouraging. With her accompany, I can keep being optimistic and survive from the stress of getting failure of researches, or from the clouded and uncertain future.

Our works in anomaly detection (Chapters 2 and 3) have been supported by NSF award DMS18-30418.

Dedication

To those who held me up over the years

Abstract

This dissertation serves as a collection of my three projects after I received the Ph.D. candidacy in 2018. The first two projects ([1] in Chapters 2 and [2] in 3, respectively), joint works with Dongmian Zou and Gilad Lerman, are about novel algorithms for unsupervised and semi-supervised anomaly detection tasks, respectively. Our new methods allow datasets with a high ratio of corruption by outliers. The third project ([3, 4] in Chapter 4), a joint work with Kshitij Tayal, Raunak Manekar, Zhong Zhuang, Vipin Kumar and Ju Sun, brings out a methodology for improving the performance of end-to-end deep learning approaches for inverse problems with many-to-one forward mappings. General features of these three projects are introduced in the following.

In Chapter 2, we propose a neural network for unsupervised anomaly detection with a novel robust subspace recovery layer (RSR layer). This layer seeks to extract the underlying subspace from a latent representation of the given data and removes outliers that lie away from this subspace. It is used within an autoencoder. The encoder maps the data into a latent space, from which the RSR layer extracts the subspace. The decoder then smoothly maps back the underlying subspace to a "manifold" close to the original inliers. Inliers and outliers are distinguished according to the distances between the original and mapped positions (small for inliers and large for outliers). Extensive numerical experiments with both image and document datasets demonstrate state-of-the-art precision and recall.

In Chapter 3, we propose a new method for novelty detection that can tolerate high corruption of the training points, whereas previous works assumed either no or very low corruption. Our method trains a robust variational autoencoder (VAE), which aims to generate a model for the uncorrupted training points. To gain robustness to high corruption, we incorporate the following four changes to the common VAE: 1. Extracting crucial features of the latent code by a carefully designed dimension reduction component for distributions; 2. Modeling the latent distribution as a mixture of Gaussian low-rank inliers and full-rank outliers, where the testing only uses the inlier model; 3. Applying the Wasserstein-1 metric for regularization, instead of the Kullback-Leibler (KL) divergence; and 4. Using a robust error for reconstruction. We establish both

robustness to outliers and suitability to low-rank modeling of the Wasserstein metric as opposed to the KL divergence. We illustrate state-of-the-art results on standard benchmarks.

In Chapter 4, we propose a methodology to resolve the irregular approximation of the inverse mapping in some inverse problems with many-to-one forward mappings; especially, we focus on 2D Fourier phase retrieval problem. In many physical systems, inputs related by intrinsic system symmetries generate the same output. So when inverting such systems, an input is mapped to multiple symmetry-related outputs. This causes fundamental difficulties for tackling these inverse problems by the emerging end-to-end deep learning approach. Taking phase retrieval as an illustrative example, we show that careful symmetry breaking on the training data can help get rid of the difficulties and significantly improve learning performance in real data experiments. We also extract and highlight the underlying mathematical principle of the proposed solution, which is directly applicable to other inverse problems.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Robust anomaly detection

Finding and utilizing patterns in data is a common task for modern machine learning systems. However, there is often some anomalous information that does not follow a common pattern and has to be recognized. For this purpose, anomaly detection aims to identify data points that "do not conform to expected behavior" [6]. We refer to such points as either anomalous or outliers. According to different tasks and scenarios of application, there are various setups for the anomaly detection problems. We follow the categorization of [5] and summarize common anomaly detection tasks in Figure 1.1 based on the availability of labels of datasets. We remark that the supervised anomaly detection task, illustrated in Figure 1.1.1, coincides with the imbalanced classification problem with two classes, which are well investigated [7, 8, 9]. Since all the training dataset is fully labeled for both the inliers and outliers, one may train a classifier and apply the trained classifier to the test dataset. However, this setting is not practical in general since in the real-world application it might be expensive or laborious to annotate datasets [10] well. In this dissertation, we developed two novel algorithms for solving unsupervised anomaly detection problems (in Chapter 2) and semi-supervised problems with corruption in the training dataset (in Chapter 3). Literature usually refer to the former task as outlier detection and the latter one as novelty detection. We will introduce these two scenarios in detail.

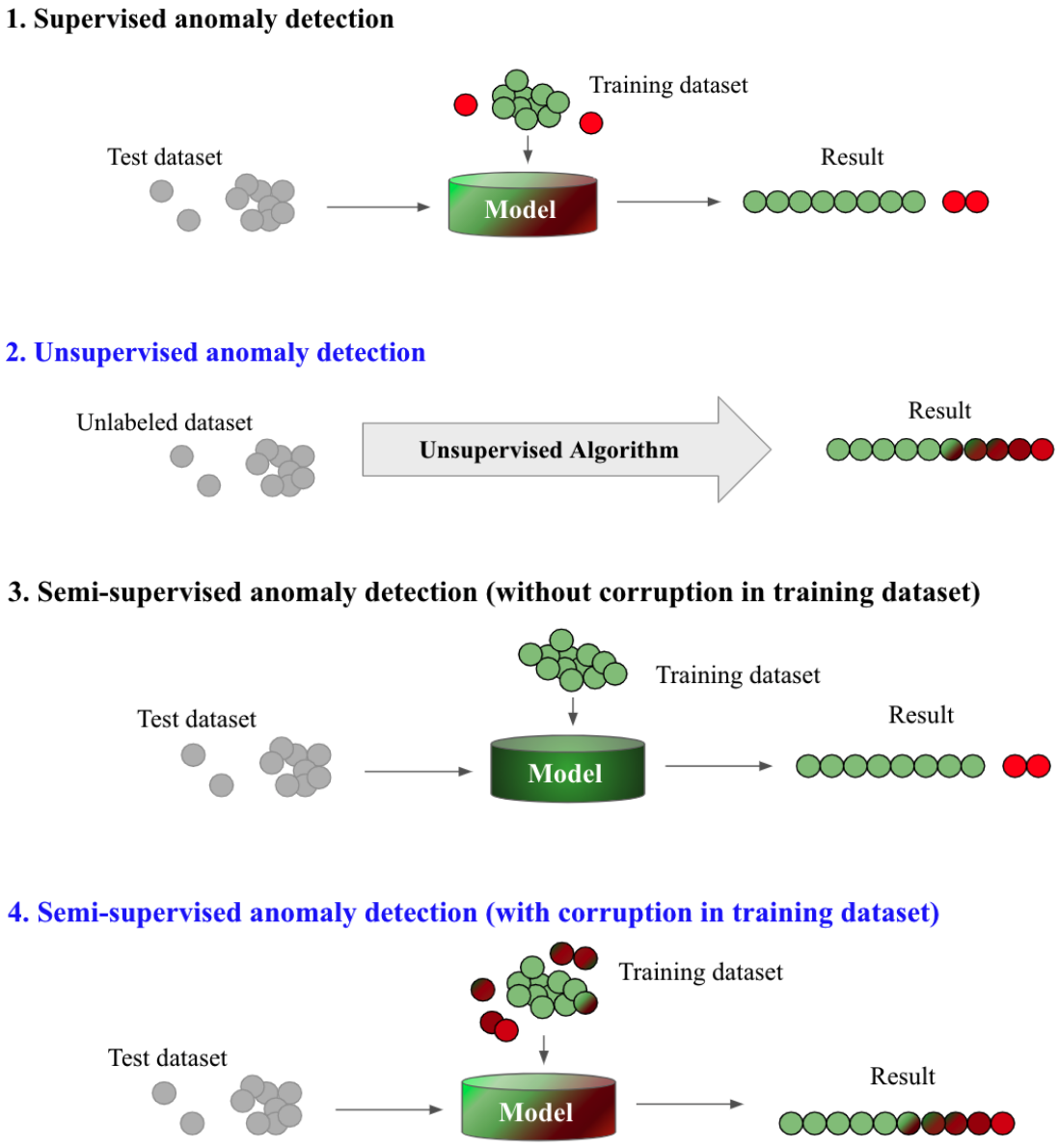


Figure 1.1: Illustration of different types of anomaly detection tasks [5]. The tasks marked in blue are addressed in this dissertation in Chapter 2 and 3, respectively.

Outlier detection (unsupervised anomaly detection) In many applications, there is no ground truth available to distinguish anomalous from normal points, and they need to be detected in an unsupervised fashion. For example, one may need to remove anomalous images from a set of images obtained by a search engine without any prior knowledge about how a normal image should look [11]. Similarly, one may need to distinguish unusual news items from a large collection of news documents without any information whether a news item is usual or not [12]. In these examples, the only assumptions are that normal data points appear more often than anomalous ones and have a simple underlying structure which is unknown to the user. In Chapter 2, we develop a new algorithm for the outlier detection which tolerates a large corruption by abnormal data points. Figure 1.1.2 illustrates the unsupervised setting anomaly detection task.

Novelty detection (semi-supervised anomaly detection) Novelty detection refers to the task of detecting testing data points that deviate from the underlying structure of a given training dataset [6, 13, 14, 15]. It finds crucial applications, in areas such as insurance and credit fraud [16], mobile robots [17] and medical diagnosis [18]. Ideally, novelty detection requires learning the underlying distribution of the training data, where sometimes it is sufficient to learn a significant feature, geometric structure or another property of the training data. One can then apply the learned distribution (or property) to detect deviating points in the test data. This is different from outlier detection [6], in which one does not have training data and has to determine the deviating points assuming that the majority of points share the same structure or properties.

We note that novelty detection is equivalent to the well-known one-class classification problem [19]. In this problem, one needs to identify members of a class in a test dataset, and consequently distinguish them from "novel" data points, given training points from this class. The points of the main class are commonly referred to as inliers and the novel ones as outliers. In Chapter 3, we consider a scenario that the training dataset has the non-trivial corruption by outliers. Our proposed method allows the training ratio of outliers per inliers up to 0.5. Figure 1.1.4 illustrates such a task which allow non-trivial corruption in the training dataset.

However, we notice that some literature use the terminology, novelty detection,

differently from ours, where it means that a training set is provided for the inliers only. Figure 1.1.3 illustrates such a task without corruption in the training dataset.

There are a myriad of solutions to novelty detection. Nevertheless, such solutions often assume that the training set is purely sampled from a single class or has few outliers. This assumption is only valid when the area of investigation has been carefully studied and there are sufficiently precise tools to collect data. However, there are different important scenarios, where this assumption does not hold. One scenario includes new studies, where it is unclear how to distinguish between normal and abnormal points. For example, in the beginning of the COVID-19 pandemic it was hard to diagnose COVID-19 patients and distinguish them from other patients with pneumonia. Another scenario occurs when it is very hard to make precise measurements, for example, when working with the highly corrupted images obtained in cryogenic electron microscopy (cryo-EM).

1.2 Breaking Symmetries of phase retrieval problems with Deep Learning

For many physical systems, we observe only the output and strive to infer the input. The inference task is captured by the umbrella term inverse problem. Formally, the underlying system is modeled by a forward mapping f, and solving the inverse problem amounts to identifying the inverse mapping f^{-1} (see Figure 1.2). Inverse problems abound in numerous fields and take diverse forms: structure from motion in computer vision [20], image restoration in image processing [21], source separation in acoustics [22], inverse scattering in physics [23], tomography in medical imaging [24], soil profile estimation in remote sensing [25], various factorization problems in machine learning [26], to name a few.

The advent of deep learning has brought tremendous novel opportunities for solving inverse problems. The most radical is perhaps the end-to-end approach: a deep neural network (DNN) is directly set up and trained to approximate the inverse mapping f^{-1}—backed by the famous universal approximation theorem [27]—based on a large set of $(\boldsymbol{x}, \boldsymbol{y})$ pairs.

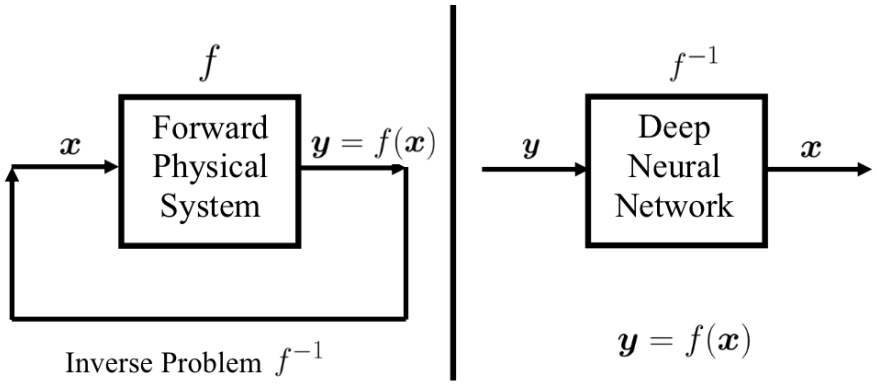


Figure 1.2: Illustration of inverse problem and the end-to-end learning approach.

Difficulty with symmetries When the forward mapping f is nonlinear and not invertible, we start to see intrinsic symmetries in many systems. To give several quick examples:

- **Blind deconvolution** [28, 29] The forward model is $\boldsymbol{y} = \boldsymbol{a} \circledast \boldsymbol{x}$, where $\boldsymbol{a}$ is the convolution kernel, $\boldsymbol{x}$ is the signal (e.g., image) of interest, and $\circledast$ denotes circular convolution. Both $\boldsymbol{a}$ and $\boldsymbol{x}$ are inputs. Here, $\boldsymbol{a} \circledast \boldsymbol{x} = (\lambda \boldsymbol{a}) \circledast (\boldsymbol{x}/\lambda)$ for any $\lambda \neq 0$, and circularly shifting $\boldsymbol{a}$ to the left and shifting $\boldsymbol{x}$ to the right by the same amount does not change $\boldsymbol{y}$.

- **Blind source separation** [22] The forward model is $\boldsymbol{Y} = \boldsymbol{A}\boldsymbol{X}$, where $\boldsymbol{A}$ is the mixing matrix and $\boldsymbol{X}$ is the source matrix and both $\boldsymbol{A}$ and $\boldsymbol{X}$ are inputs. The scaling symmetry similar to above is also present here. Moreover, signed permutations are another kind of symmetries, i.e., $\boldsymbol{A}\boldsymbol{X} = (\boldsymbol{A}\boldsymbol{\Pi}\boldsymbol{\Sigma})\big(\boldsymbol{\Sigma}^{-1}\boldsymbol{\Pi}^{-1}\boldsymbol{X}\big)$ for any permutation matrix $\boldsymbol{\Pi}$ and any diagonal sign matrix $\boldsymbol{\Sigma}$. We note that for both blind deconvolution and blind source separation, depending on structures of the inputs, there may be other symmetries that we have not covered here. The symmetries we have discussed tend to be persistent.

- **Fourier phase retrieval** [30] The forward model is $\boldsymbol{Y} = |\mathscr{F}(\boldsymbol{X})|^2$, where $\boldsymbol{X} \in \mathbb{C}^n$ and $\boldsymbol{Y} \in \mathbb{R}^m$ are matrices and $\mathscr{F}$ is a 1D oversampled Fourier matrix. The operation $|\cdot|$ takes complex magnitudes of the entries elementwise. It is known that translations and conjugate flippings applied on $\boldsymbol{X}$, and also global phase transfer of the form $e^{i\theta}\boldsymbol{X}$ all lead to the same $\boldsymbol{Y}$.

Solving these inverse problems means recovering the input up to the intrinsic system symmetries, as evidently this is the best one can hope for. However, symmetries can cause significant difficulty for the end-to-end approach. In Chapter 4 we elaborate problems raised by the intrinsic symmetries of the systems. Furthermore, we focus on the phase retrieval problems and proposal a general methodology to preprocess the training data points to "break the symmetry".

1.3 Structure of the dissertation

The outline of the dissertation is structured as the following.

- Chapter 2 introduces a novel model, RSRAE, for the unsupervised anomaly detection task which allows a large corruption for the dataset.
- Chapter 3 introduces a novel model, MAW, for the semi-supervised anomaly detection task which tolerates an non-trivial corruption for the training dataset. The testing dataset can also contain a large portion of corruptions.
- Chapter 4 introduces a general methodology to resolve the irregular approximation of the inverse mapping in Fourier phase retrieval problems which has the many-to-one forward mappings (so one-to-many mapping for the "inverse").
- Chapter 5 briefly summarizes the works in Chapter 2, 3 and 4. It is followed by the discussion of several potential future works.
- Chapter A provides supplementary proofs for propositions in Chapter 2.
- Chapter B provides supplementary proofs for propositions in Chapter 3.
- Chapter C provides supplementary proofs for propositions in Chapter 4.
- Chapter D describes the metrics used for anomaly detection in Chapters 2 and 3 and the measurement used for Fourier phase retrieval problems in Chapter 4.
- Chapter E summarizes numerical results presented in Chapter 3 as tables.

Chapter 2

Robust Subspace Recovery Layer for Unsupervised Anomaly Detection

2.1 Introduction

Some early methods for anomaly detection relied on Principal Component Analysis (PCA) [31]. Here one assumes that the underlying unknown structure of the normal samples is linear. However, PCA is sensitive to outliers and will often not succeed in recovering the linear structure or identifying the outliers [32, 33]. More recent ideas of Robust PCA (RPCA) [34, 33] have been considered for some specific problems of anomaly detection or removal [35, 36]. RPCA assumes sparse corruption, that is, few elements of the data matrix are corrupted. This assumption is natural for some special problems in computer vision, in particular, background subtraction [37, 34, 33]. However, a natural setting of anomaly detection with hidden linear structure may assume instead that a large portion of the data points are fully corrupted. The mathematical framework that addresses this setting is referred to as robust subspace recovery (RSR) [32].

While Robust PCA and RSR try to extract linear structure or identify outliers lying away from such structure, the underlying geometric structure of many real datasets is

nonlinear. Therefore, one needs to extract crucial features of the nonlinear structure of the data while being robust to outliers. In order to achieve this goal, we propose to use an autoencoder (composed of an encoder and a decoder) with an RSR layer. We refer to it as RSRAE (RSR autoencoder). It aims to robustly and nonlinearly reduce the dimension of the data in the following way. The encoder maps the data into a high-dimensional space. The RSR layer linearly maps the embedded points into a low-dimensional subspace that aims to learn the hidden linear structure of the embedded normal points. The decoder maps the points from this subspace to the original space. It aims to map the normal points near their original locations, and the anomalous points far from their original locations.

Ideally, the encoder maps the normal data to a linear space and any anomalies lie away from this subspace. In this ideal scenario, anomalies can be removed by an RSR method directly applied to the data embedded by the encoder. Since the linear model for the normal data embedded by the encoder is only approximate, we do not directly apply RSR to the embedded data. Instead, we minimize a sum of the reconstruction error of the autoencoder and the RSR error for the data embedded by the encoder. We advocate for an alternating procedure, so that the parameters of the autoencoder and the RSR layer are optimized in turn.

2.1.1 Structure of this chapter

Section 2.2 reviews works that are directly related to the proposed RSRAE and highlights the original contributions of this work. Section 2.3 explains the proposed RSRAE, and in particular, its RSR layer and total energy function. Section 2.4 illustrates the mechanism of RSRAE with an artificial example. Section 2.5 includes extensive experimental evidence demonstrating effectiveness of RSRAE with both image and document data. Section 2.6 further compares RSRAE with other robust methods such as vanilla RSR, robust PCA and its variants. Section 2.7 tests the sensitivity of RSRAE to hyperparameters. Section 2.8 mathematically relates the linear autoencoders with the subspace problem and further proves a relationship between the proposed RSR loss with Wasserstein distance.

2.2 Related Works and Contribution

We review related works in Section 2.2.1 and highlight our contribution in Section 2.2.2.

2.2.1 Related Works

Several recent works have used autoencoders for anomaly detection. [11] proposed the earliest work on anomaly detection via an autoencoder, while utilizing large reconstruction error of outliers. They apply an iterative and cyclic scheme, where in each iteration, they determine the inliers and use them for updating the parameters of the autoencoder. [38] apply ℓ_2 normalization for the latent code of the autoencoder and also consider the case of multiple modes for the normal samples. Instead of using the reconstruction error, they apply k-means clustering for the latent code, and identify outliers as points whose latent representations are far from all the cluster centers. [39] also use an autoencoder with clustered latent code, but they fit a Gaussian Mixture Model using an additional neural network. Restricted Boltzmann Machines (RBMs) are similar to autoencoders. [40] define "energy functions" for RBMs that are similar to the reconstruction losses for autoencoders. They identify anomalous samples according to large energy values. [41] propose using ideas of RPCA within an autoencoder, where they alternatively optimize the parameters of the autoencoder and a sparse residual matrix.

The above works are designed for datasets with a small fraction of outliers. However, when this fraction increases, outliers are often not distinguished by high reconstruction errors or low similarity scores. In order to identify them, additional assumptions on the structure of the normal data need to be incorporated. For example, [35] decompose the input data into two parts: low-rank and sparse (or column-sparse). The low-rank part is fed into an autoencoder and the sparse part is imposed as a penalty term with the ℓ_1-norm (or $\ell_{2,1}$-norm for column-sparsity).

In this work, we use a term analogous to the $\ell_{2,1}$-norm, which can be interpreted as the sum of absolute deviations from a latent subspace. However, we do not decompose the data a priori, but minimize an energy combining this term and the reconstruction error. Minimization of the former term is known as least absolute deviations in RSR [32]. It was first suggested for RSR and related problems in [42, 43, 44]. The robustness to outliers of this energy, or of relaxed versions of it, was studied in [45, 46, 47, 48,

49, 50, 51]. In particular, [51] established its well-behaved landscape under special, though natural, deterministic conditions. Under similar conditions, they guaranteed fast subspace recovery by a simple algorithm that aims to minimize this energy.

Another directly related idea for extracting useful latent features is an addition of a linear self-expressive layer to an autoencoder [52]. It is used in the different setting of unsupervised subspace clustering. By imposing the self-expressiveness, the autoencoder is robust to an increasing number of clusters. Although self-expressiveness also improves robustness to noise and outliers, [52] aims at clustering and thus its goal is different than ours. Furthermore, their self-expressive energy does not explicitly consider robustness, while ours does. [53] consider a somewhat parallel idea of imposing a loss function to increase the robustness of representation. However, their goal is to increase the margin between classes and their method only applies to a supervised setting in anomaly detection, where the normal data is multi-modal.

2.2.2 Contribution of this work

This work introduces an RSR layer within an autoencoder. It incorporates a special regularizer that enforces an outliers-robust linear structure in the embedding obtained by the encoder. We clarify that the method does not alternate between application of the autoencoder and the RSR layer, but fully integrates these two components. Our experiments demonstrate that a simple incorporation of a "robust loss" within a regular autoencoder does not work well for anomaly detection. We try to explain this and also the improvement obtained by incorporating an additional RSR layer.

Our proposed architecture is simple to implement. Furthermore, the RSR layer is not limited to a specific design of RSRAE but can be put into any well-designed autoencoder structure. The epoch time of the proposed algorithm is comparable to those of other common autoencoders. Furthermore, our experiments show that RSRAE competitively performs in unsupervised anomaly detection tasks.

RSRAE addresses the unsupervised setting, but is not designed to be highly competitive in the semi-supervised or supervised settings, where one has access to training data from the normal class or from both classes, respectively. In these settings, RSRAE functions like a regular autoencoder without taking an advantage of its RSR layer, unless the training data for the normal class is corrupted with outliers.

The use of RSR is not restricted to autoencoders. We establish some preliminary analysis for RSR within a generative adversarial network (GAN) [54, 55] in Section 2.8. More precisely, we show that a linear WGAN intrinsically incorporates RSR in some special settings, although it is unclear how to impose an RSR layer.

2.3 The structure of RSRAE

In Section 2.3.1, we motivate and overview the model of RSRAE. In Section 2.3.2, we provide the detailed algorithms of implementing RSRAE and its variance RSRAE+.

Figure 2.1 illustrates the general idea of RSRAE and can help with understanding the scheme of it.

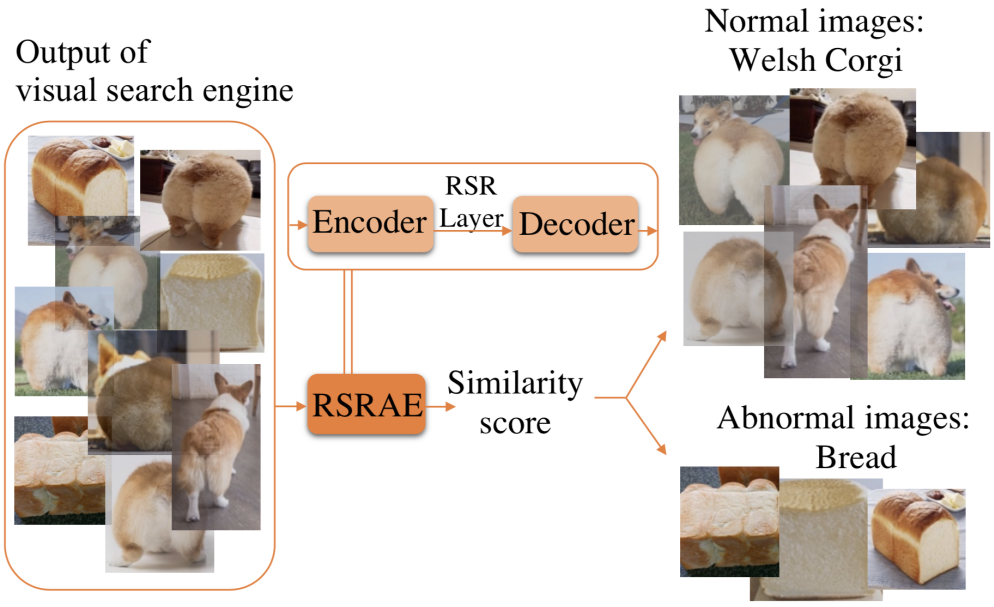


Figure 2.1: Demonstration of RSRAE for anomaly detection using a set of images obtained by a visual search engine.

2.3.1 RSR layer for outlier removal

We assume input data $\{\boldsymbol{x}^{(t)}\}_{t=1}^{N}$ in $\mathbb{R}^M$, and denote by $\boldsymbol{X}$ its corresponding data matrix, whose t-th column is $\boldsymbol{x}^{(t)}$. The encoder of RSRAE, $\mathscr{E}$, is a neural network that maps each data point, $\boldsymbol{x}^{(t)}$, to its latent code $\boldsymbol{z}^{(t)} = \mathscr{E}(\boldsymbol{x}^{(t)}) \in \mathbb{R}^D$. The RSR layer is a linear

transformation $\boldsymbol{A} \in \mathbb{R}^{d\times D}$ that reduces the dimension to d. That is, $\tilde{\boldsymbol{z}}^{(t)} = \boldsymbol{A}\boldsymbol{z}^{(t)} \in \mathbb{R}^d$. The decoder $\mathscr{D}$ is a neural network that maps $\tilde{\boldsymbol{z}}^{(t)}$ to $\tilde{\boldsymbol{x}}^{(t)}$ in the original ambient space $\mathbb{R}^M$.

We can write the forward maps in a compact form using the corresponding data matrices as follows:

$$\boldsymbol{Z} = \mathscr{E}(\boldsymbol{X}), \quad \tilde{\boldsymbol{Z}} = \boldsymbol{A}\boldsymbol{Z}, \quad \tilde{\boldsymbol{X}} = \mathscr{D}(\tilde{\boldsymbol{Z}}). \tag{2.1}$$

Ideally, we would like to optimize RSRAE so it only maintains the underlying structure of the normal data. We assume that the original normal data lies on a d-dimensional "manifold" in $\mathbb{R}^D$ and thus the RSR layer embeds its latent code into $\mathbb{R}^d$. In this ideal optimization setting, the similarity between the input and the output of RSRAE is large whenever the input is normal and small whenever the input is anomalous. Therefore, by thresholding a similarity measure, one may distinguish between normal and anomalous data points.

In practice, the matrix $\boldsymbol{A}$ and the parameters of $\mathscr{E}$ and $\mathscr{D}$ are obtained by minimizing a loss function, which is a sum of two parts: the reconstruction loss from the autoencoder and the loss from the RSR layer. For $p > 0$, an $\ell_{2,p}$ reconstruction loss for the autoencoder is

$$L^p_{\mathrm{AE}}(\mathscr{E}, \boldsymbol{A}, \mathscr{D}) = \sum_{t=1}^{N} \left\| \boldsymbol{x}^{(t)} - \tilde{\boldsymbol{x}}^{(t)} \right\|_2^p . \tag{2.2}$$

In order to motivate our choice of RSR loss, we review a common formulation for the original RSR problem. In this problem one needs to recover a linear subspace, or equivalently an orthogonal projection $\boldsymbol{P}$ onto this subspace. Assume a dataset $\{\boldsymbol{y}^{(t)}\}_{t=1}^N$ and let $\boldsymbol{I}$ denote the identity matrix in the ambient space of the dataset. The goal is to find an orthogonal projector $\boldsymbol{P}$ of dimension d whose subspace robustly approximates this dataset. The least q-th power deviations formulation for $q > 0$, or least absolute deviations when $q = 1$ [32], seeks $\boldsymbol{P}$ that minimizes

$$\hat{L}(\boldsymbol{P}) = \sum_{t=1}^{N} \left\| (\boldsymbol{I} - \boldsymbol{P})\, \boldsymbol{y}^{(t)} \right\|_2^q . \tag{2.3}$$

The solution of this problem is robust to some outliers when $q \leq 1$ [47, 50]; furthermore, $q < 1$ can result in a wealth of local minima and thus $q = 1$ is preferable [47, 50].

A similar loss function to (2.3) for RSRAE is

$$\begin{aligned} L^q_{\mathrm{RSR}}(\boldsymbol{A}) &= \lambda_1 L_{\mathrm{RSR}_1}(\boldsymbol{A}) + \lambda_2 L_{\mathrm{RSR}_2}(\boldsymbol{A}) \\ &:= \lambda_1 \sum_{t=1}^{N} \left\| \boldsymbol{z}^{(t)} - \boldsymbol{A}^{\mathrm{T}} \underbrace{\boldsymbol{A}\boldsymbol{z}^{(t)}}_{\tilde{\boldsymbol{z}}^{(t)}} \right\|_2^q + \lambda_2 \left\| \boldsymbol{A}\boldsymbol{A}^{\mathrm{T}} - \boldsymbol{I}_d \right\|_{\mathrm{F}}^2 , \end{aligned} \tag{2.4}$$

where $\boldsymbol{A}^{\mathrm{T}}$ denotes the transpose of $\boldsymbol{A}$, $\boldsymbol{I}_d$ denotes the $d \times d$ identity matrix and $\|\cdot\|_{\mathrm{F}}$ denotes the Frobenius norm. Here $\lambda_1, \lambda_2 > 0$ are predetermined hyperparameters, though we later show that one may solve the underlying problem without using them. We note that the first term in the weighted sum of (2.4) is close to (2.3) as long as $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}$ is close to an orthogonal projector. To enforce this requirement we introduced the second term in the weighted sum of (2.4). In Section 2.8.4 we discuss further properties of the RSR energy and its minimization.

To emphasize the effect of outlier removal, we take $p = 1$ in (2.2) and $q = 1$ in (2.4). That is, we use the $l_{2,1}$ norm, or the formulation of least absolute deviations, for both reconstruction and RSR. The loss function of RSRAE is the sum of the two loss terms in (2.2) and (2.4), that is,

$$L_{\mathrm{RSRAE}}(\mathscr{E}, \boldsymbol{A}, \mathscr{D}) = L^1_{\mathrm{AE}}(\mathscr{E}, \boldsymbol{A}, \mathscr{D}) + L^1_{\mathrm{RSR}}(\boldsymbol{A}). \tag{2.5}$$

We remark that the sole minimization of L^1_{AE}, without L^1_{RSR}, is not effective for anomaly detection. We numerically demonstrate this in Section 2.5.5 and also try to explain it in Section 2.8.1.

Our proposed algorithm for optimizing (2.5), which we refer to as the RSRAE algorithm, uses alternating minimization. It iteratively backpropagates the three terms L^1_{AE}, L_{RSR_1}, L_{RSR_2} and accordingly updates the parameters of the RSR autoencoder. For clarity, we describe this basic procedure in Algorithm 1 of Section 2.3.2. It is independent of the values of the parameters λ_1 and λ_2. Note that the additional gradient step with respect to the RSR loss just updates the parameters in $\boldsymbol{A}$. Therefore it does not significantly increase the epoch time of a standard autoencoder for anomaly detection. Another possible method, which we refer to as RSRAE+, is direct minimization of L_{RSRAE} with predetermined λ_1 and λ_2 via auto-differentiation (see Algorithm 2 of

Section 2.3.2). Section 2.5.5 demonstrates that in general, RSRAE performs better than RSRAE+, though it is possible that similar performance can be achieved by carefully tuning the parameters λ_1 and λ_2 when implementing RSRAE+.

We remark that a standard autoencoder is obtained by minimizing only L^2_{AE}, without the RSR loss. One might hope that minimizing L^1_{AE} may introduce the needed robustness. However, Section 2.5.5 demonstrates that results obtained by minimizing L^1_{AE} or L^2_{AE} are comparable, and are worse than those of RSRAE and RSRAE+.

2.3.2 Details of RSRAE and RSRAE+

The implementations of both RSRAE and RSRAE+ are simple. For completeness we provide here their details in algorithm boxes. The codes are publicly available in `https://github.com/dmzou/RSRAE`. Algorithm 1 describes RSRAE, which minimizes (2.5) by alternating minimization. It denotes the vectors of parameters of the encoder and decoder by $\boldsymbol{\theta}$ and $\boldsymbol{\varphi}$, respectively.

We clarify some guidelines for choosing default parameters, which we follow in all reported experiments. We set ϵ_{AE}, $\epsilon_{\mathrm{RSR}_1}$ and $\epsilon_{\mathrm{RSR}_2}$ to be zero. In general, we use networks with dense layers but for image data we use convolutional layers. We prefer using tanh as the activation function due to its smoothness. However, for a dataset that does not lie in the unit cube, we use either a ReLU function if all of its coordinates are positive, or a leaky ReLU function otherwise. The network parameters and the elements of $\boldsymbol{A}$ are initialized to be i.i.d. standard normal. In all numerical experiments, we set the number of columns of $\boldsymbol{A}$ to be 10, that is, $d = 10$. The learning rate is chosen so that there is a sufficient improvement of the loss values after each epoch. Instead of fixing ϵ_{T}, we report the AUC and AP scores for different values of ϵ_{T}.

Algorithm 2 describes RSRAE+, which minimizes (2.5) with fixed λ_1 and λ_2 by auto-differentiation.

Algorithm 1 RSRAE

Input: Data $\{\boldsymbol{x}^{(t)}\}_{t=1}^{N}$; thresholds $\boldsymbol{\epsilon}_{\mathrm{AE}}$, $\boldsymbol{\epsilon}_{\mathrm{RSR}_1}$, $\boldsymbol{\epsilon}_{\mathrm{RSR}_2}$, $\boldsymbol{\epsilon}_{\mathrm{T}}$; architecture and initial parameters of $\mathscr{E}$, $\mathscr{D}$, $\boldsymbol{A}$ (including number of columns of $\boldsymbol{A}$); number of epochs & batches; learning rate for backpropagation; similarity measure

Output: Labels of data points as normal or anomalous

for each epoch **do**
  Divide input data into batches
  for each batch **do**
    if $L^1_{\mathrm{AE}}(\boldsymbol{\theta}, \boldsymbol{A}, \boldsymbol{\varphi}) > \boldsymbol{\epsilon}_{\mathrm{AE}}$ **then**
      Backpropagate $L^1_{\mathrm{AE}}(\boldsymbol{\theta}, \boldsymbol{A}, \boldsymbol{\varphi})$ w.r.t. $\boldsymbol{\theta}, \boldsymbol{A}, \boldsymbol{\varphi}$ & update $\boldsymbol{\theta}, \boldsymbol{A}, \boldsymbol{\varphi}$
    end if
    if $L^1_{\mathrm{RSR}_1}(\boldsymbol{A}) > \boldsymbol{\epsilon}_{\mathrm{RSR}_1}$ **then**
      Backpropagate $L^1_{\mathrm{RSR}_1}(\boldsymbol{A})$ w.r.t. $\boldsymbol{A}$ & update $\boldsymbol{A}$
    end if
    if $L^1_{\mathrm{RSR}_2}(\boldsymbol{A}) > \boldsymbol{\epsilon}_{\mathrm{RSR}_2}$ **then**
      Backpropagate $L^1_{\mathrm{RSR}_2}(\boldsymbol{A})$ w.r.t. $\boldsymbol{A}$ & update $\boldsymbol{A}$
    end if
  end for
end for
for $t = 1, \ldots, N$ **do**
  Calculate similarity between $\boldsymbol{x}^{(t)}$ and $\tilde{\boldsymbol{x}}^{(t)}$
  if similarity $\geq \boldsymbol{\epsilon}_{\mathrm{T}}$ **then**
    $\boldsymbol{x}^{(t)}$ is normal
  else
    $\boldsymbol{x}^{(t)}$ is anomalous
  end if
end for
return Normality labels for $t = 1, \ldots, N$

2.4 Demonstration of RSRAE for artificial data

For illustrating the performance of RSRAE, in comparison with a regular autoencoder, we consider a simple artificial geometric example. We assume corrupted data whose normal part is embedded in a "Swiss roll manifold"[1], which is a two-dimensional manifold in $\mathbb{R}^3$. More precisely, the normal part is obtained by mapping 1,000 points uniformly

[1] `https://scikit-learn.org/stable/modules/generated/sklearn.datasets.make_swiss_roll.html`

Algorithm 2 RSRAE+

Input: Data $\{\boldsymbol{x}^{(t)}\}_{t=1}^{N}$; thresholds $\boldsymbol{\epsilon}_{\mathrm{AE}}$, $\boldsymbol{\epsilon}_{\mathrm{T}}$; architecture and initial parameters of $\mathscr{E}$, $\mathscr{D}$, $\boldsymbol{A}$ (including number of columns of $\boldsymbol{A}$); parameters of the the energy function λ_1, λ_2; number of epochs & batches; learning rate for backpropagation; similarity measure

Output: Labels of data points as normal or anomalous

for each epoch **do**
  Divide input data into batches
  for each batch **do**
    if $L^1_{\mathrm{AE}}(\boldsymbol{\theta}, \boldsymbol{A}, \boldsymbol{\varphi}) > \boldsymbol{\epsilon}_{\mathrm{AE}}$ **then**
      Backpropagate $L^1_{\mathrm{AE}}(\boldsymbol{\theta}, \boldsymbol{A}, \boldsymbol{\varphi}) + \lambda_1 L^1_{\mathrm{RSR}_1}(\boldsymbol{A}) + \lambda_2 L^1_{\mathrm{RSR}_2}(\boldsymbol{A})$ w.r.t. $\boldsymbol{\theta}, \boldsymbol{A}, \boldsymbol{\varphi}$ & update $\boldsymbol{\theta}, \boldsymbol{A}, \boldsymbol{\varphi}$
    end if
  end for
end for
for $t = 1, \ldots, N$ **do**
  Calculate similarity between $\boldsymbol{x}^{(t)}$ and $\tilde{\boldsymbol{x}}^{(t)}$
  if similarity $\geq \boldsymbol{\epsilon}_{\mathrm{T}}$ **then**
    $\boldsymbol{x}^{(t)}$ is normal
  else
    $\boldsymbol{x}^{(t)}$ is anomalous
  end if
end for
return Normality labels for $t = 1, \ldots, N$

sampled from the rectangle $[3\pi/2, 9\pi/2] \times [0, 21]$ into $\mathbb{R}^3$ by the function

$$(s, t) \mapsto (t\cos(t), s, t\sin(t)). \tag{2.6}$$

The anomalous part is obtained by i.i.d. sampling of 500 points from an isotropic Gaussian distribution in $\mathbb{R}^3$ with zero mean and standard deviation 2 in any direction. Figure 2.2a illustrates such a sample, where the inliers are in black and the outliers are in blue. We remark that Fig 2.3a is identical.

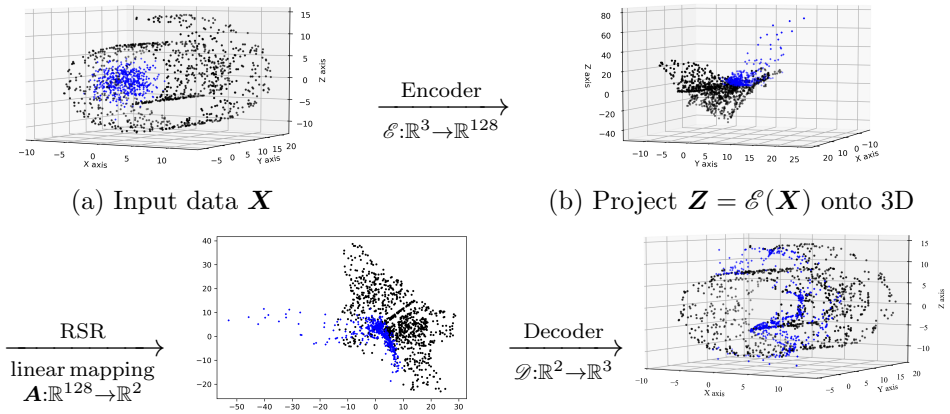
We construct the RSRAE with the following structure. The encoder is composed of fully-connected layers of sizes (32, 64, 128). The decoder is composed of fully connected layers of sizes (128, 64, 32, 3). Each fully connected layer is activated by the leaky ReLU function with $\alpha = 0.2$. The intrinsic dimension for the RSR layer, that, is the

number of columns of $\boldsymbol{A}$, is $d = 2$.

For comparison, we construct the regular autoencoder AE (see Section 2.5.5). Recall that both of them have the same architecture (including the linear map $\boldsymbol{A}$), but AE minimizes the ℓ_2 loss function in (2.7) (with $p = 2$) without an additional RSR loss. We optimize both models with 10,000 epochs and a batch gradient descent using Adam [56] with a learning rate of 0.01.

The reconstructed data ($\tilde{\boldsymbol{X}}$) using RSRAE and AE are plotted in Figures 2.2d and 2.3d, respectively. We further demonstrate the output obtained by the encoder and the RSR layer. The output of the encoder, $\boldsymbol{Z} = \mathscr{E}(\boldsymbol{X})$, lies in $\mathbb{R}^{128}$. For visualization purposes we project it onto a $\mathbb{R}^3$ as follows. We first find two vectors that span the image of $\boldsymbol{A}$ and we add to it the "principal direction" of $\boldsymbol{Z}$ orthogonal to the span of $\boldsymbol{A}$. We project $\boldsymbol{Z}$ onto the span of these 3 vectors. Figures 2.2b and 2.3b show these projections for RSRAE and AE, respectively. Figures 2.2c and 2.3c demonstrate the respective mappings of $\boldsymbol{Z}$ by $\boldsymbol{A}$ during the RSR layer.

Figures 2.2d and 2.3d imply that the set of reconstructed normal points in RSRAE seem to lie on the original manifold, whereas the reconstructed normal points by AE seem to only lie near, but often not on the Swiss roll manifold. More importantly, the anomalous points reconstructed by RSRAE seem to be sufficiently far from the set of original anomalous points, unlike the reconstructed points by AE. Therefore, RSRAE can better distinguish anomalies using the distance between the original and reconstructed points, where small values are obtained for normal points and large ones for anomalous ones. Figure 2.4 demonstrates this claim. They plot the histograms of the distance between the original and reconstructed points when applying RSRAE and AE, where distances for normal and anomalous points are distinguished by color. Clearly, RSRAE distinguishes normal and anomalous data better than AE.



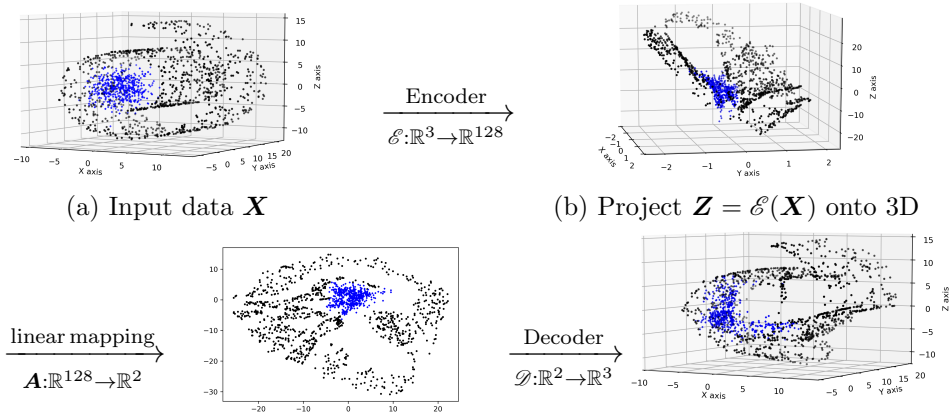
(a) Input data $\boldsymbol{X}$

(b) Project $\boldsymbol{Z} = \mathscr{E}(\boldsymbol{X})$ onto 3D

(c) $\tilde{\boldsymbol{Z}} = \boldsymbol{A}\boldsymbol{Z}$

(d) Output of RSRAE $\tilde{\boldsymbol{X}} = \mathscr{D}(\tilde{\boldsymbol{Z}})$

Figure 2.2: Demonstration of the output of the encoder, RSR layer and decoder of RSRAE on a corrupted Swiss roll dataset.



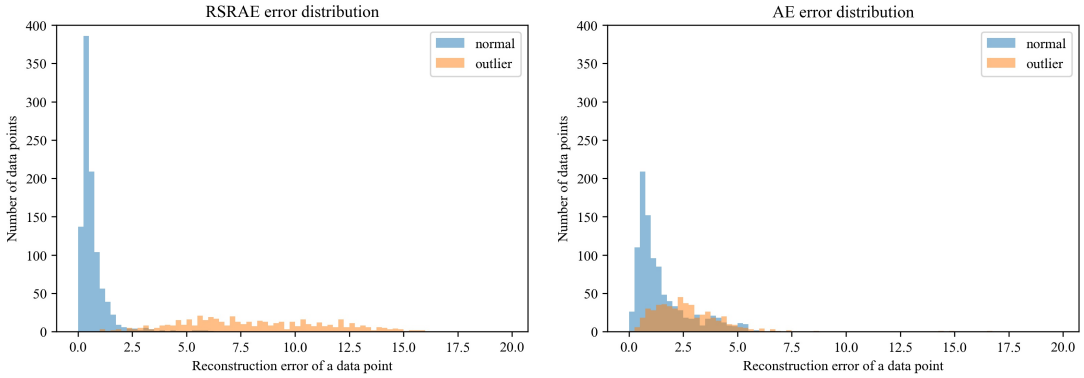
(a) Input data $\boldsymbol{X}$

(b) Project $\boldsymbol{Z} = \mathscr{E}(\boldsymbol{X})$ onto 3D

(c) $\tilde{\boldsymbol{Z}} = \boldsymbol{A}\boldsymbol{Z}$

(d) Output of AE $\tilde{\boldsymbol{X}} = \mathscr{D}(\tilde{\boldsymbol{Z}})$

Figure 2.3: Demonstration of the output of the encoder, mapping by $\boldsymbol{A}$, and decoder of AE on a corrupted Swiss roll dataset.



(a) Error distribution for RSRAE.

(b) Error distribution for AE.

Figure 2.4: Demonstration of the reconstruction error distribution for RSRAE and AE.

2.5 Experimental Results

In Section 2.5.1 we introduce the implemented datasets. In Section 2.5.2, we describe compared benchmarks. In Section 2.5.3 we describe the experiment setting of RSRAE in detail. In Section 2.5.4, we demonstrate the performance of RSRAE compared to the benchmarks. In Section 2.5.5, we test RSRAE with its variants.

2.5.1 Datasets

We test our method [2]on five datasets: Caltech 101 [57], Fashion-MNIST [58], Tiny Imagenet (a small subset of Imagenet [59]), Reuters-21578 [60] and 20 Newsgroups [61].

Caltech 101 contains 9,146 RGB images labeled according to 101 distinct object categories. We take the 11 categories that contain at least 100 images and randomly choose 100 images per category. We preprocess all 1100 images to have size $32 \times 32 \times 3$ and pixel values normalized between -1 and 1. In each experiment, the inliers are the 100 images from a certain category and we sample $c \times 100$ outliers from the rest of 1000 images of other categories, where $c \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$.

Fashion-MNIST contains 28×28 grayscale images of clothing and accessories, which are categorized into 10 classes. We use the test set which contains 10,000 images and normalize pixel values to lie in $[-1, 1]$. In each experiment, we fix a class and the

[2]Our implementation is available at `https://github.com/dmzou/RSRAE.git`

inliers are the test images in this class. We randomly sample $c \times 1{,}000$ outliers from the rest of classes (here and below c is as above). Since there are around 1000 test images in each class, the outlier ratio is approximately c.

Tiny Imagenet contains 200 classes of RGB images from a distinct subset of Imagenet. We select 10 classes with 500 training images per class. We preprocess the images to have size $32 \times 32 \times 3$ and pixel values in $[-1, 1]$. We further represent the images by deep features obtained by a ResNet [62] with dimension 256 In each experiment, 500 inliers are from a fixed class and $c \times 500$ outliers are from the rest of classes.

Reuters-21578 contains 90 text categories with multi-labels. We consider the five largest classes with single labels and randomly sample from them 360 documents per class. The documents are preprocessed into vectors of size 26,147 by sequentially applying the TFIDF transformer and Hashing vectorizer [63]. In each experiment, the inliers are the documents of a fixed class and $c \times 360$ outliers are randomly sampled from the other classes.

20 Newsgroups contains newsgroup documents with 20 different labels. We sample 360 documents per class and preprocess them as above into vectors of size 10,000. In each experiment, the inliers are the documents from a fixed class and $c \times 360$ outliers are sampled from the other classes.

2.5.2 Benchmarks

We compare RSRAE with the following benchmarks: Local Outlier Factor (LOF) [64], One-Class SVM (OCSVM) [65, 66], Isolation Forest (IF) [67], Deep Structured Energy Based Models (DSEBMs) [40], Geometric Transformations (GT) [68], and Deep Autoencoding Gaussian Mixture Model (DAGMM) [39]. We briefly describe these methods below.

Local Outlier Factor (LOF) measures the local deviation of a given data point with respect to its neighbors. If the LOF of a data point is too large then the point is determined to be an outlier.

One-Class SVM (OCSVM) learns a margin for a class of data. Since outliers contribute less than the normal class, it also applies to the unsupervised setting [5]. It is usually applied with a non-linear kernel.

Isolation Forest (IF) determines outliers by looking at the number of splittings

needed for isolating a sample. It constructs random decision trees. A short path length for separating a data point implies a higher probability that the point is an outlier.

Geometric Transformations (GT) applies a variety of geometric transforms to input images and consequently creates a self-labeled dataset, where the labels are the types of transformations. Its anomaly detection is based on Dirichlet Normality score according to the softmax output from a classification network for the labels.

Deep Structured Energy-Based Models (DSEBMs) outputs an energy function which is the negative log probability that a sample follows the data distribution. The energy based model is connected to an autoencoder to avoid the need of complex sampling methods.

Deep Autoencoding Gaussian Mixture Model (DAGMM) is also a deep autoencoder model. It optimizes an end-to-end structure that contains both an autoencoder and an estimator for Gaussian Mixture Model. The anomaly detection is done after modeling the density function of the Gaussian Mixture Model.

Of those benchmarks, LOF, OCSVM and IF are traditional, while powerful methods, for unsupervised anomaly detection and do not involve neural networks. DSEBMs, DAGMM and GT are more recent and all involve neural networks. DSEBMs is built for unsupervised anomaly detection. DAGMM and GT are designed for semi-supervised anomaly detection, but allow corruption. We use them to learn a model for the inliers and assign anomaly scores using the combined set of both inliers and outliers. GT only applies to image data. We implemented DSEBMs, DAGMM and GT using the codes[3] from [68] with minimal modification so that they adapt to the data described above and the available GPUs in our machine. The LOF, OCSVM and IF methods are adapted from the scikit-learn packages.

2.5.3 Settings of the experiment

We describe the structure of the RSRAE as follows. For the image datasets without deep features, the encoder consists of three convolutional layers: 5×5 kernels with 32 output channels, strides 2; 5×5 kernels with 64 output channels, strides 2; and 3×3 kernels with 128 output channels, strides 2. The output of the encoder is flattened and the RSR layer transforms it into a 10-dimensional vector. That is, we fix $d = 10$

[3] `https://github.com/izikgo/AnomalyDetectionTransformations`

in all experiments. The decoder consists of a dense layer that maps the output of the RSR layer into a vector of the same shape as the output of the encoder, and three deconvolutional layers: 3×3 kernels with 64 output channels, strides 2; 5×5 kernels with 32 output channels, strides 2; 5×5 kernels with 1 (grayscale) or 3 (RGB) output channels, strides 2. For the preprocessed document datasets or the deep features of Tiny Imagenet, the encoder is a fully connected network with size (32, 64, 128), the RSR layer linearly maps the output of the encoder to dimension 10, and the decoder is a fully connected network with size (128, 64, 32, D) where D is the dimension of the input. Batch normalization is applied to each layer of the encoders and the decoders. The output of the RSR layer is ℓ_2-normalized before applying the decoder. For DSEBMs and DAGMM we use the same number of layers and the same dimensions in each layer for the autoencoder as in RSRAE. For each experiment, the RSRAE model is optimized with Adam using a learning rate of 0.00025 and 200 epochs. The batch size is 128 for each gradient step. The setting of training is consistent for all the neural network based methods.

The two main hyperparameters of RSRAE are the intrinsic dimension d and learning rate. Their values were fixed above. Section 2.7.1 - Section 2.7.3 demonstrates stability to changes in these values.

All experiments were executed on a Linux machine with 64GB RAM and four GTX1080Ti GPUs. For all experiments with neural networks, we used TensorFlow and Keras. We report runtimes in Section 2.7.4.

2.5.4 Results

We summarize the precision and recall of our experiments by the AUC (area under curve) and AP (average precision) scores. For completeness, we include the definitions of these common scores in Appendix D.1. We compute them by considering the outliers as "positive". We remark that we did not record the precision-recall-F1 scores, as in [11, 39], since in practice it requires knowledge of the outlier ratio.

Figures. 2.5 and 2.6 present the AUC and AP scores of RSRAE and the methods described in Section 2.5.2 for the datasets described above, where GT is only applied to image data without deep features. For each constant c (the outlier ratio) and each

method, we average the AUC and AP scores over 5 runs with different random initializations and also compute the standard deviations. For brevity of presentation, we report the averaged scores among all classes and designate the averaged standard deviations by bars.

The results indicates that RSRAE clearly outperforms other methods in most cases, especially when c is large. Indeed, the RSR layer was designed to handle large outlier ratios. For Fashion MNIST and Tiny Imagenet with deep features, IF performs similarly to RSRAE, but IF performs poorly on the document datasets. OCSVM is the closest to RSRAE for the document datasets but it is generally not so competitive for the image datasets.

2.5.5 Comparison with Variations of RSRAE

We use one image dataset (Caltech 101) and one document dataset (Reuters-21578) and compare between RSRAE and three variations of it. The first one is RSRAE+ (see Section 2.3) with $\lambda_1 = \lambda_2 = 0.1$ in (2.4) (these parameters were optimized on 20 Newsgroup, though results with other choices of parameters are later demonstrated in Section 2.7.3). The next two are simpler autoencoders without RSR layers: AE-1 minimizes L^1_{AE}, the $\ell_{2,1}$ reconstruction loss; and AE minimizes L^2_{AE}, the $\ell_{2,2}$ reconstruction loss (it is a regular autoencoder for anomaly detection). We maintain the same architecture as that of RSRAE, including the matrix $\boldsymbol{A}$, but use different loss functions.

Figures. 2.7 and 2.8 report the AUC and AP scores. We see that for the two datasets RSRAE+ with the prespecified λ_1 and λ_2 does not perform as well as RSRAE, but its performance is still better than AE and AE-1. This is expected since we chose λ_1 and λ_2 after few trials with a different dataset, whereas RSRAE is independent of these parameters. The performance of AE and AE-1 is clearly worse, and they are also not as good as some methods compared with in Section 2.5.4. At last, AE is generally comparable with AE-1.

2.6 Comparison with vanilla RSR and RCAE

We demonstrate basic properties of our framework by comparing it to two different frameworks. The first framework is direct RSR, which tries to model the inliers by a

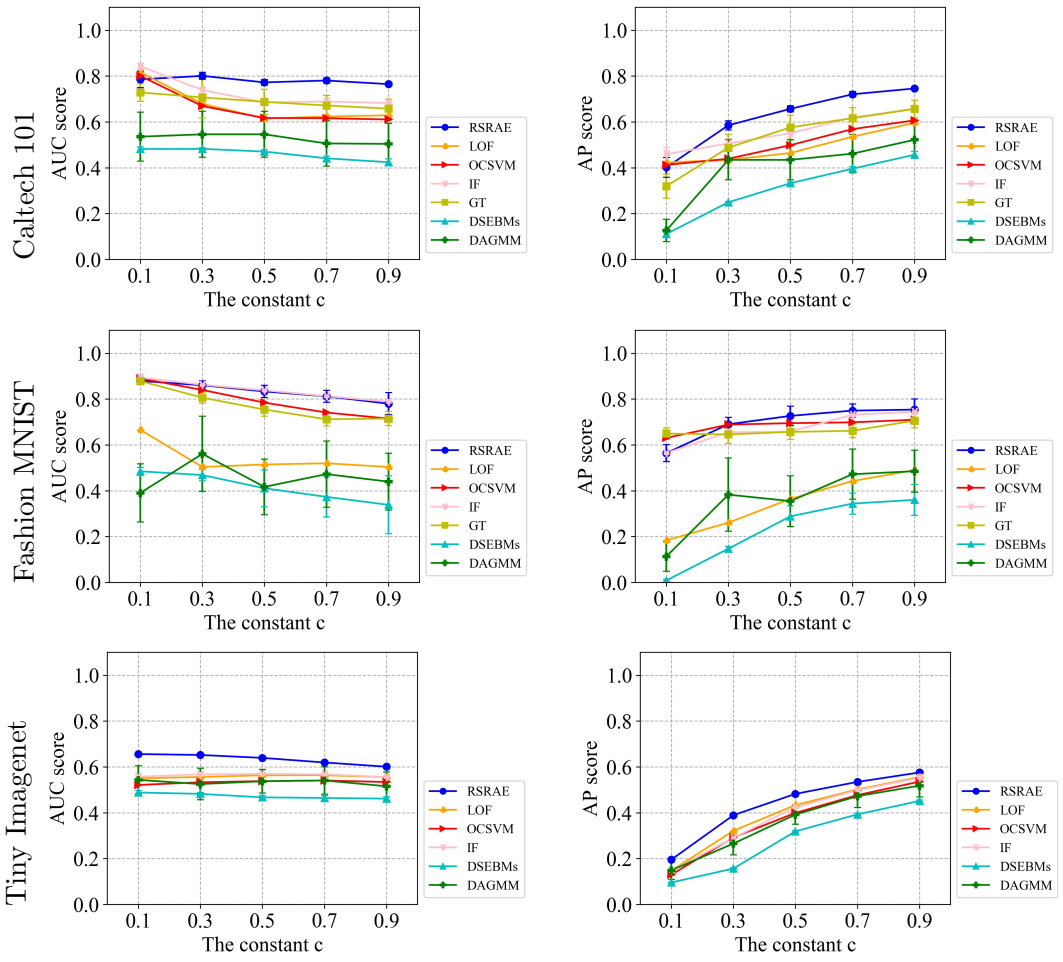


Figure 2.5: AUC and AP scores for RSRAE using Caltech 101, Fashion MNIST and Tiny Imagenet.

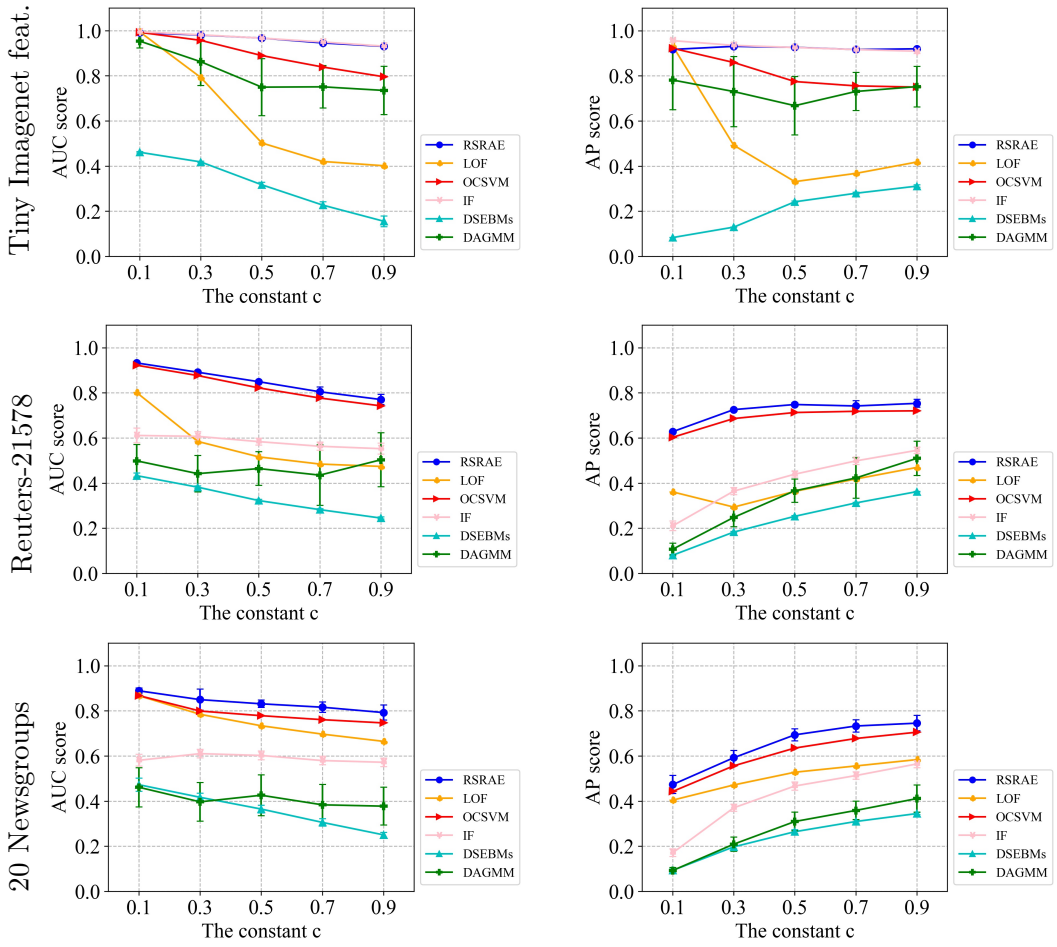


Figure 2.6: AUC and AP scores for RSRAE using Tiny Imagenet with deep features, Reuters-21578 and 20 Newsgroups.

low-dimensional subspace, as opposed to the nonlinear model discussed in here. Based on careful comparison of RSR methods in [32], we use the Fast Median Subspace (FMS) algorithm [50] and its normalized version, the Spherical FMS (SFMS). The other framework can be viewed a nonlinear version of RPCA, instead of RSR. It assumes sparse elementwise corruption of the data matrix, instead of corruption of whole data points, or equivalently, of some columns of the data matrix. For this purpose we use the Robust Convolutional Autoencoder (RCAE) algorithm of [41], who advocate it as "extension of robust PCA to allow for a nonlinear manifold that explains most of the data". We

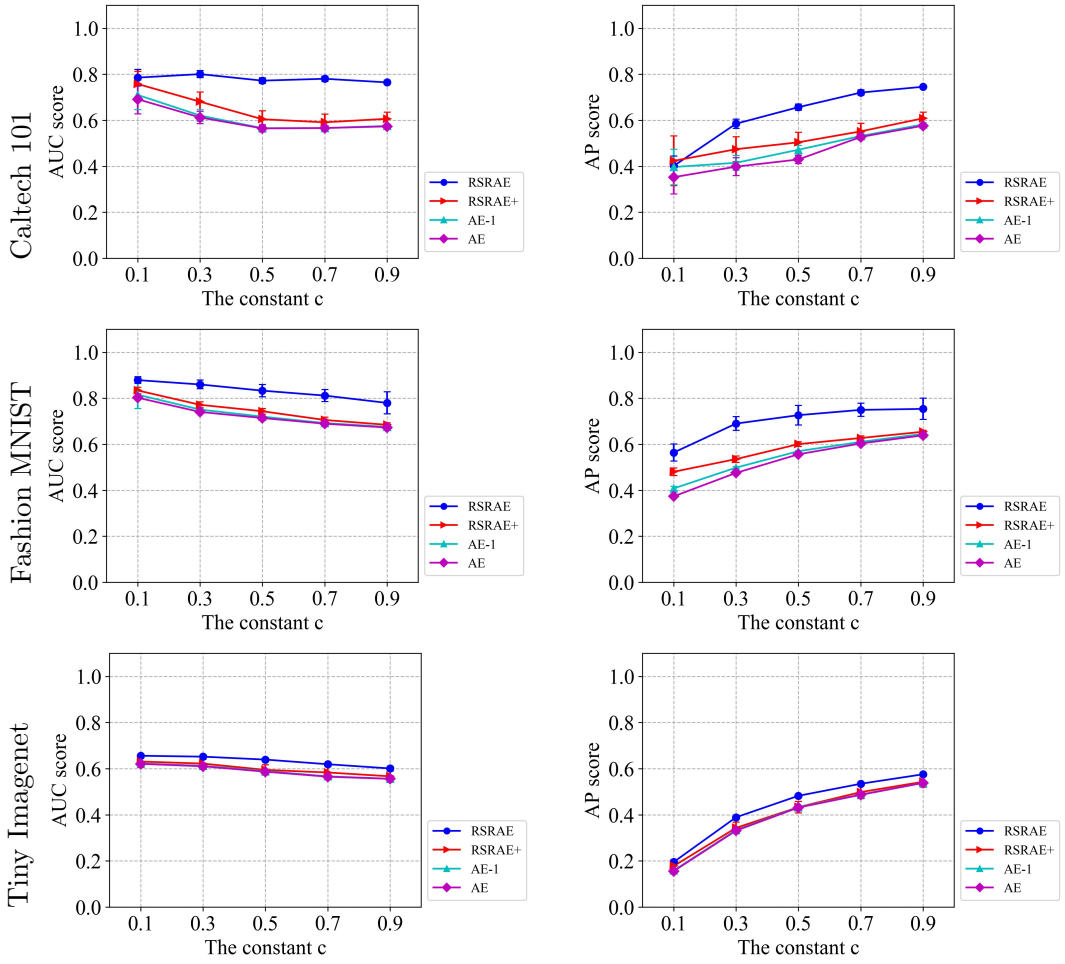


Figure 2.7: AUC and AP scores for RSRAE and alternative formulations using Caltech 101, Fashion MNIST and Tiny Imagenet.

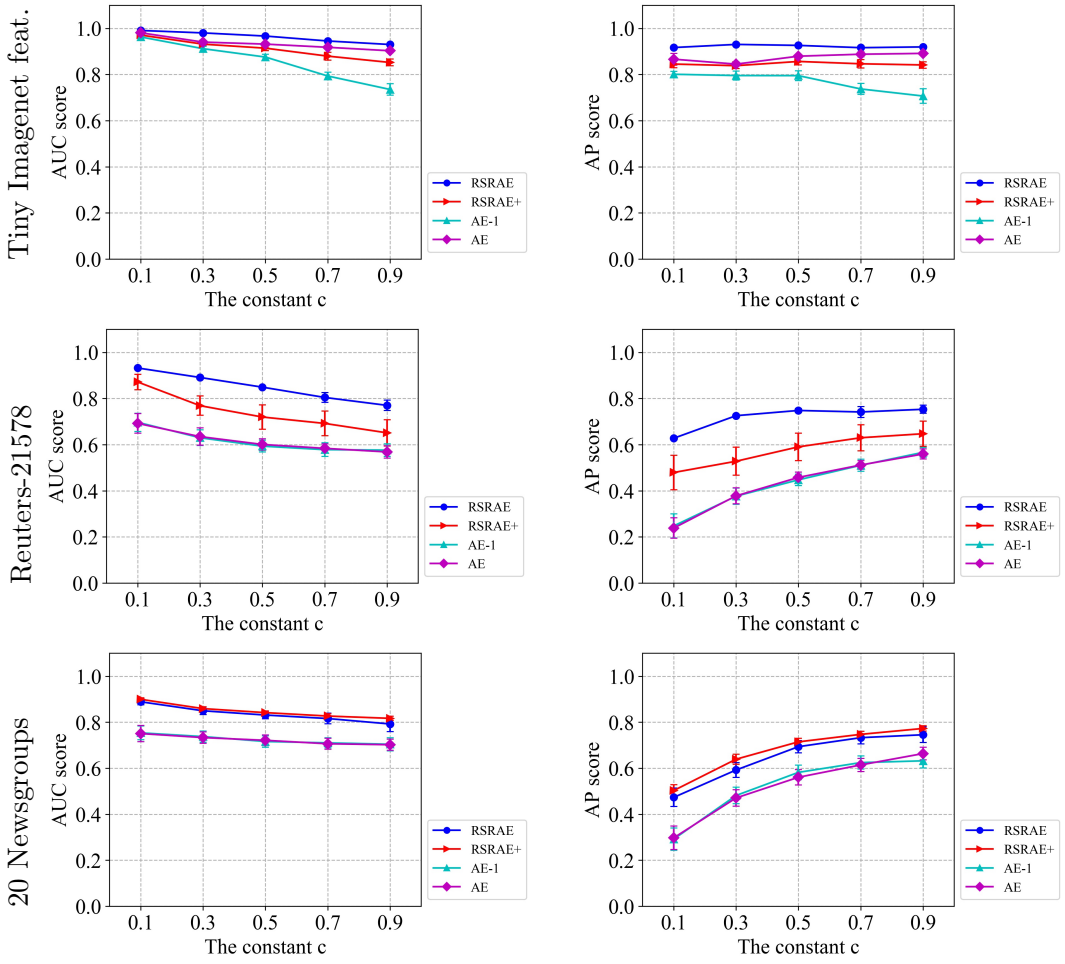


Figure 2.8: AUC and AP scores for RSRAE and alternative formulations using deep features of Tiny Imagenet, Reuters-21578 and 20 Newsgroup.

adopt the same network structures as in Section 2.5.2.

Figure 2.9 reports comparisons of RSRAE, FMS, SFMS and RCAE on the datasets used in Section 2.5.4. We first note that both FMS and SFMS are not effective for the datasets we have been using. That is, the inliers in these datasets are not well-approximated by a linear model. It is also interesting to notice that without normalization to the sphere, FMS can be much worse than SFMS. That is, SFMS is often way more robust to outliers than FMS. This observation and the fact that there are no obvious normalization procedures a general autoencoder (see Section 2.8) clarifies why the mere use of the L^1_{AE} loss for an autoencoder is not expected to be robust enough to outliers.

Comparing with RSRAE, we note that RCAE is not a competitive method for these datasets. This is not surprising since the model of RCAE, which assumes sparse elementwise corruption, does not fit well to the problem of anomaly detection, but to other problems, such as background detection.

2.7 Sensitivity to hyperparameters and runtime comparison

We examine the sensitivity of some of the reported results to changes in the hyperparameters. Section 2.7.1 tests the sensitivity of RSRAE to changes in the intrinsic dimension d. Section 2.7.2 tests the sensitivity of RSRAE to changes in the learning rate. Section 2.7.3 tests the sensitivity of RSRAE+ to changes in λ_1 and λ_2. Section 2.7.4 compare the runtime between RSRAE and benchmarks.

2.7.1 Sensitivity to the intrinsic dimension

In the experiments reported in Section 2.5 we fixed $d = 10$. Here we check the sensitivity of the reported results to changes in d. We use the same datasets of Section 2.5.4 with an outlier ratio of $c = 0.5$ and test the following values of d: $1, 2, 5, 8, 10, 12, 15, 20, 30, 40, 50$. Figure 2.10 reports the AUC and AP scores for these choice of d and for these datasets with $c = 0.5$. We note that, in general, our results are not sensitive to choices of $d \leq 30$.

We believe that the structure of these datasets is complex, and is not represented by a smooth manifold of a fixed dimension. Therefore, low-dimensional encoding of the

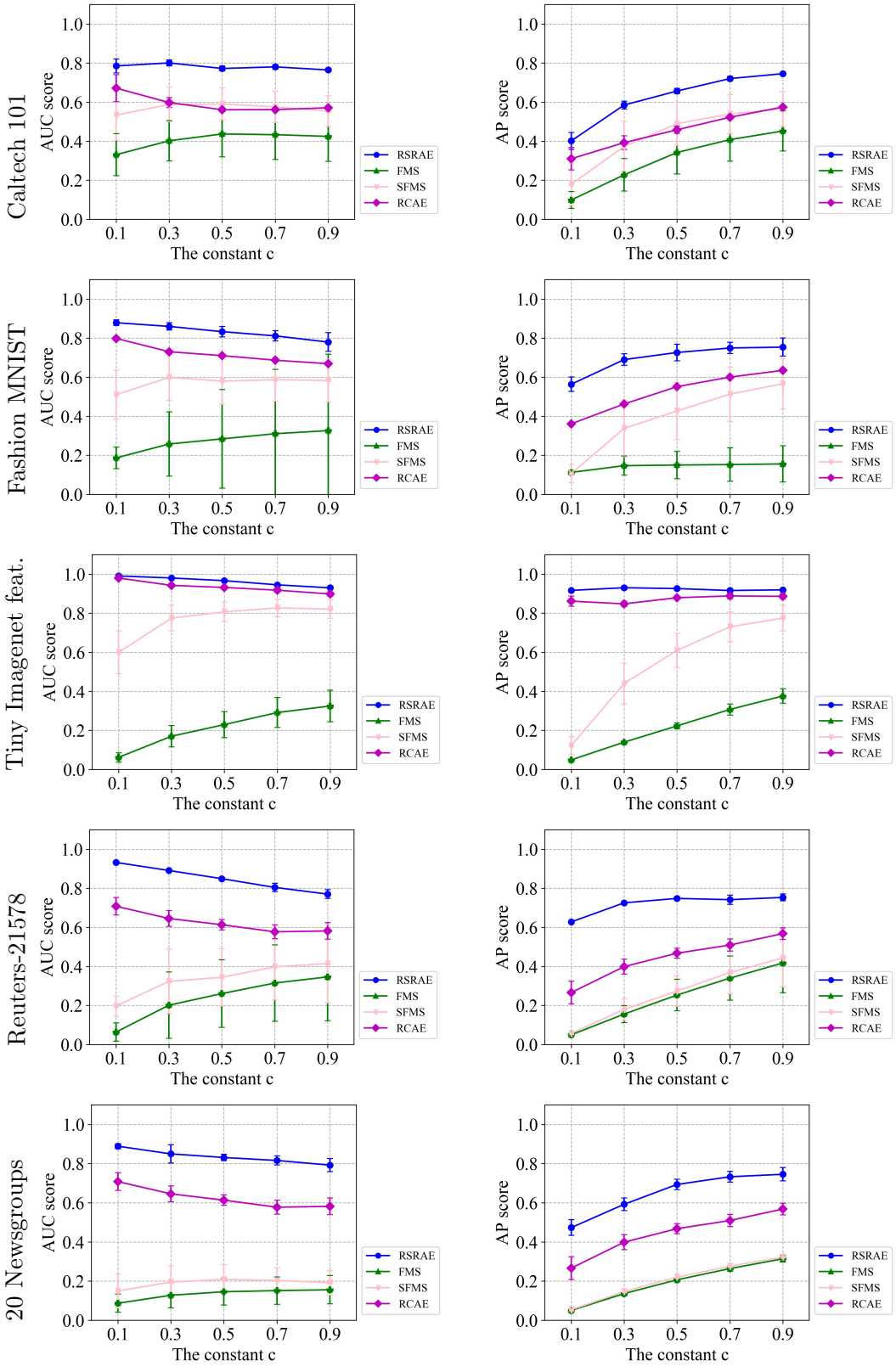


Figure 2.9: AUC and AP scores for RSRAE, FMS, SFMS and RCAE using Caltech 101, Fashion MNIST, Tiny Imagenet with deep features, Reuters-21578 and 20 Newsgroups.

inliers is beneficial with various choices of low dimensions.

When d gets closer to D the performance deteriorates. Such a decrease in accuracy is noticeable for Reuters-21578 and 20 Newsgroups, where for both datasets $D = 128$. For the image data sets (without deep features) $D = 1152$ and thus only relatively small values of d were tested. As an example of large d for an image dataset, we consider the case of $d = D = 1152$ in Caltech101 with $c = 0.5$. In this case, AUC = 0.619 and AP = 0.512, which are very low scores.

We conclude that in our experiments (with $c = 0.5$), RSRAE was stable in d around our choice of $d = 10$.

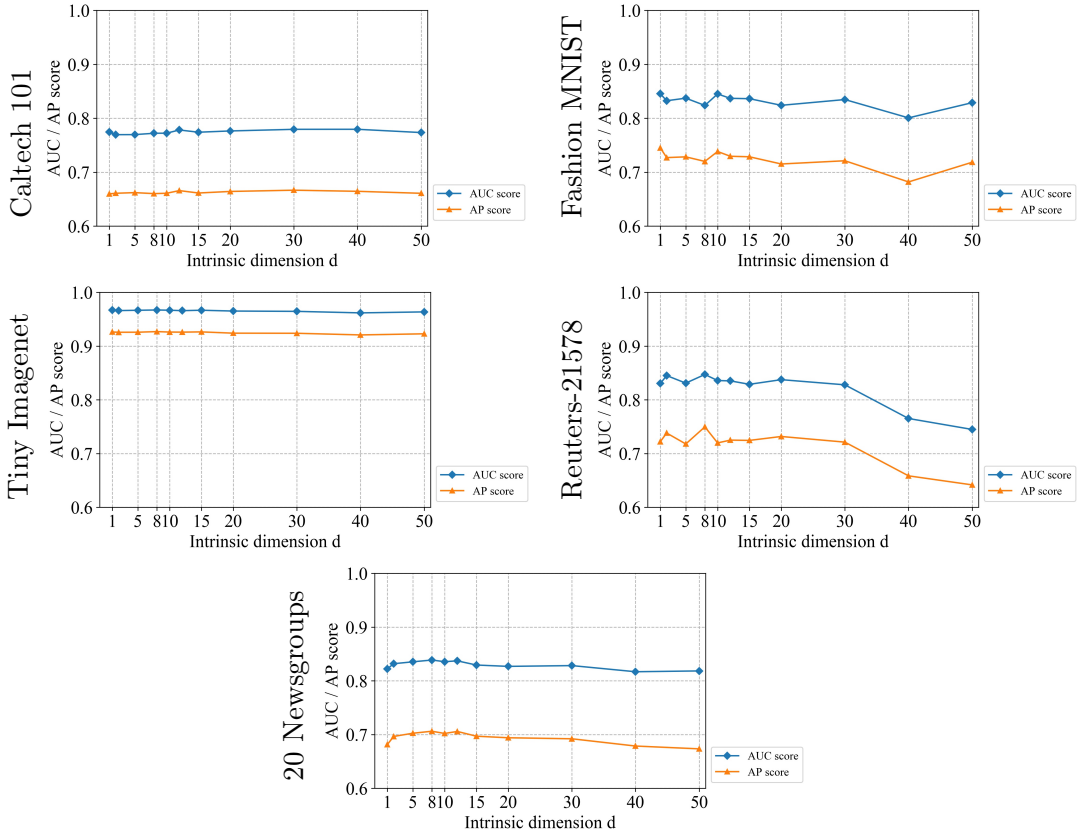


Figure 2.10: AUC and AP scores for different choices of d. The datasets are the same as those in Section 2.5.4, where the outlier ratio is $c = 0.5$.

2.7.2 Sensitivity to the learning rate

In the experiments reported in Section 2.5 we fixed the learning rate for RSRAE to be 0.00025. Here we check the sensitivity of the reported results to changes in the learning rate. We use the same datasets of Section 2.5.4 with an outlier ratio of $c = 0.5$ and test the following values of the learning rate: 0.0001, 0.00025, 0.0005, 0.001, 0.0025, 0.005, 0.01, 0.025, 0.05, 0.1. Figure. 2.11 reports the AUC and AP scores for these values and for these datasets (with $c = 0.5$). We note that the performance is stable for learning rates not exceeding 0.01.

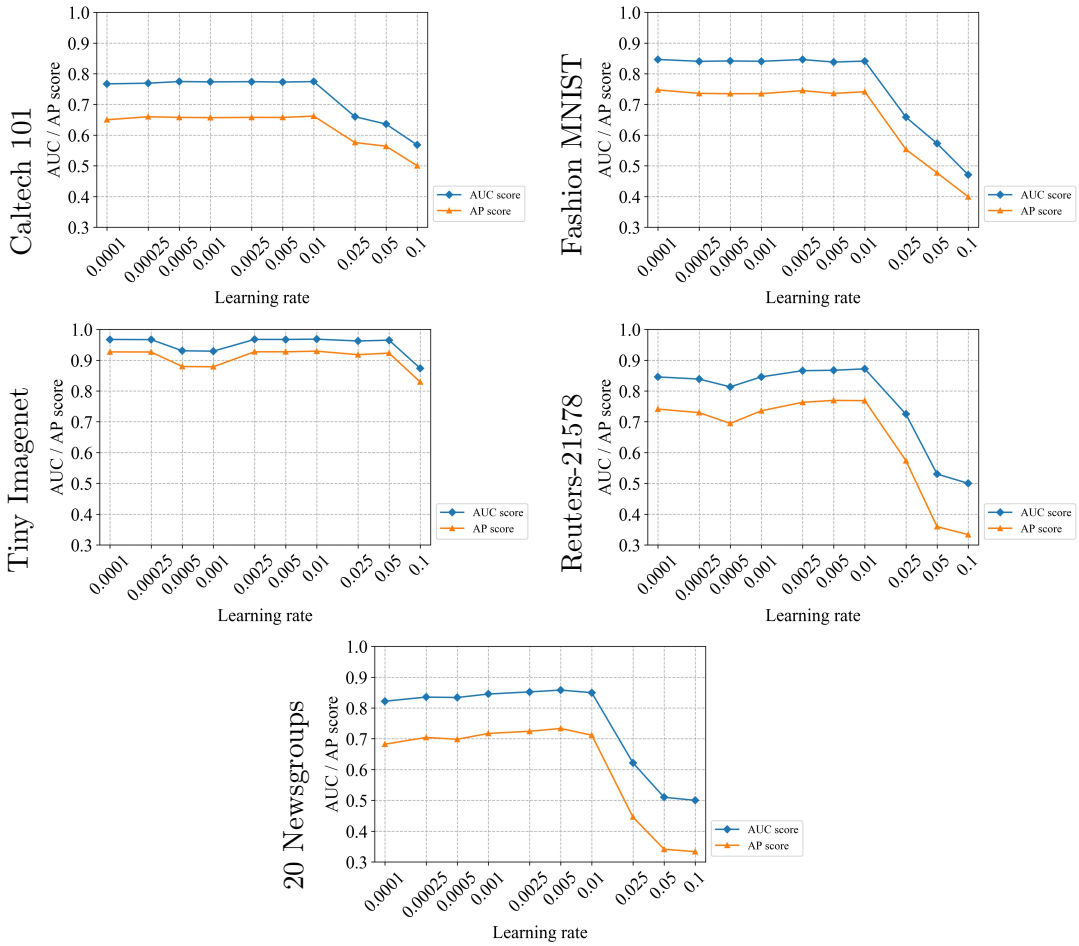


Figure 2.11: AUC and AP scores for various learning rates. The datasets are the same as those in Section 2.5.4, where the outlier ratio is $c = 0.5$.

2.7.3 Sensitivity of RSRAE+ to λ_1 and λ_2

We study the sensitivity of RSRAE+ to different choices of λ_1 and λ_2. We recall that RSRAE does not require these parameters. It is still interesting to check such sensitivity and find out whether careful tuning of these parameters in RSRAE+ can yield better scores than those of RSRAE. We use the same datasets of Section 2.5.4 with an outlier ratio of $c = 0.5$ and simultaneously test the following values of either λ_1 or λ_2: $0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1.0, 2.0$. Figures 2.12 and 2.13 report the AUC and AP scores for these values and datasets (with $c = 0.5$). For each subfigure, the above values of λ_1 and λ_2 are recorded on the x and y axes, respectively. The darker colors of the heat map correspond to larger scores. For comparison, the corresponding AUC or AP score of RSRAE is indicated in the title of each subfigure.

We note that RSRAE+ is more sensitive to λ_1 than λ_2. Furthermore, as λ_1 increases the scores are often more stable to changes in λ_1. That is, the magnitudes of the derivatives of the scores with respect to λ_1 seem to generally decrease with λ_1. In Section 2.5.5 we used $\lambda_1 = \lambda_2 = 0.1$ as this choice seemed optimal for the independent set of 20 Newsgroup. We note though that optimal hyperparameters depend on the dataset and it is thus not a good idea to optimize them using different datasets. They also depend on the choice of c, but for brevity we only test them with $c = 0.5$.

At last we note that the AUC and AP scores of RSRAE are comparable to the fine-tuned ones of RSRAE+ (where $c = 0.5$). We thus advocate using the alternating minimization of RSRAE, which is independent of λ_1 and λ_2.

2.7.4 Runtime comparison

Table 2.1 records runtimes for all the methods and datasets in Section 2.5.4 with the choice of $c = 0.5$. More precisely, a runtime is the the time needed to complete a single experiment, where 200 epoches were used for the neural networks. The table averages each runtime over the different classes.

Note that LOF, OCSVM and IF are faster than the rest of methods since they do not require training neural networks. We also note that the runtime of RSRAE is competitive in comparison to the other tested methods, that is, DSEBMs, DAGMM, and GT. The neural network structures of these four methods are the same, and thus

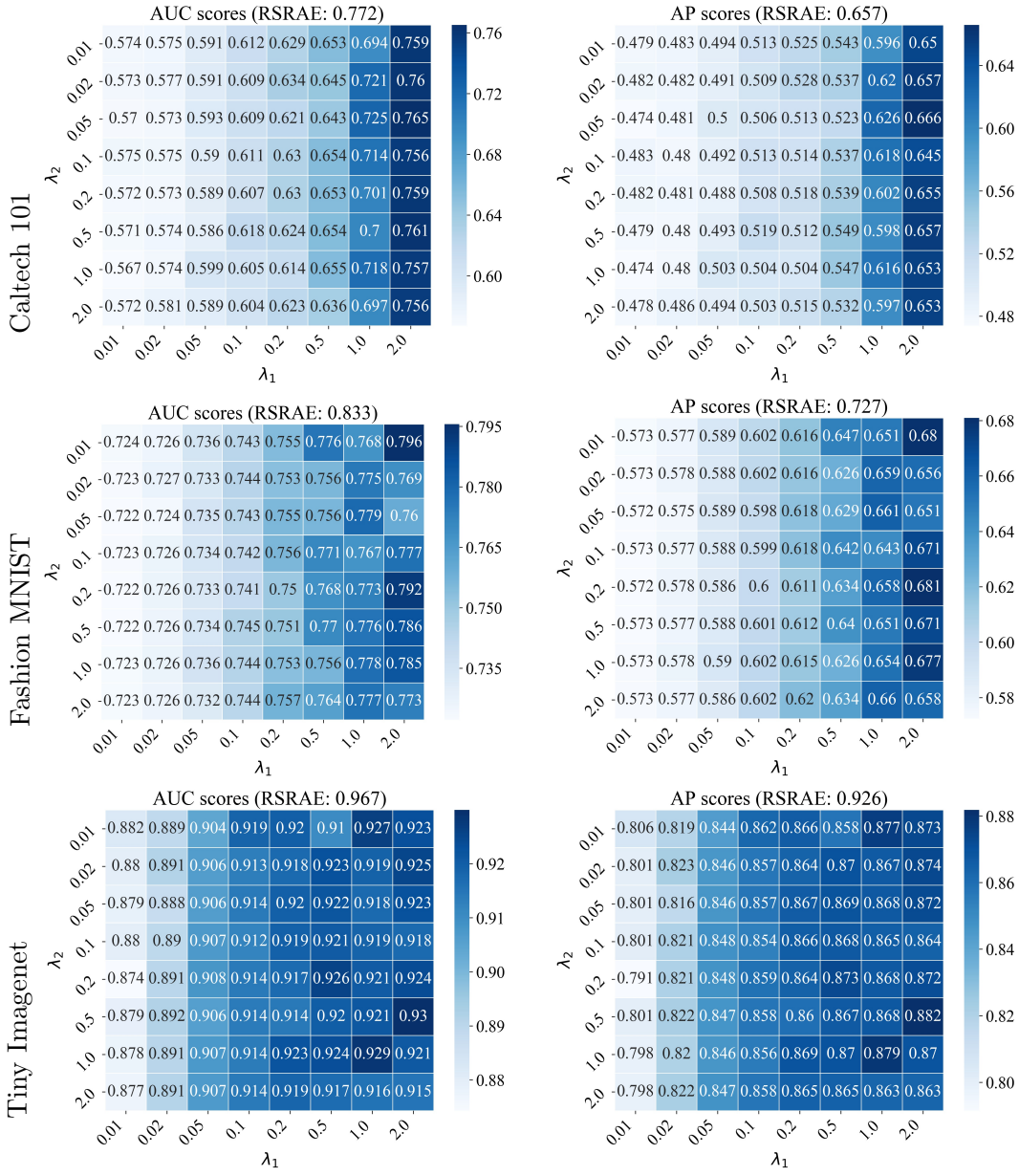


Figure 2.12: AUC and AP scores for RSRAE+ with various choices of λ_1 and λ_2 for Caltech 101, Fashion MNIST and Tiny Imagenet with deep features, where $c = 0.5$.

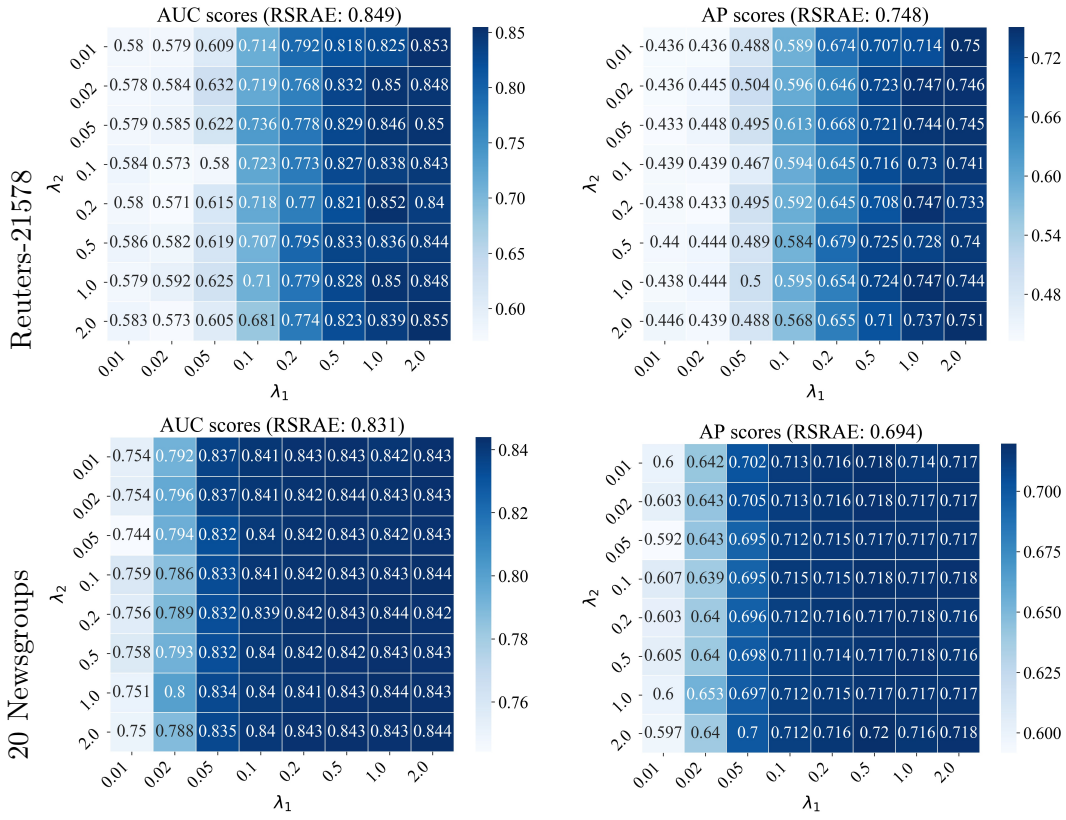


Figure 2.13: AUC and AP scores for RSRAE+ with various choices of λ_1 and λ_2 using Reuters-21578 and 20 Newsgroup, where $c = 0.5$.

the difference in runtime is mainly due to different pre and post processing.

Since GT was only applied to the image datasets without deep features, its runtime is not available (N/A) for the last three datasets.

Table 2.1: Runtime comparison (in seconds) are reported for all methods and datasets in Section 2.5.4, where the outlier ratio is $c = 0.5$.

Benchmarks \ Datasets	Caltech 101	Fashion MNIST	Tiny Imagenet	Reuters-21578	20 Newsgroups
LOF	0.233	7.163	0.707	25.342	10.516
OCSVM	0.120	3.151	0.473	8.726	4.169
IF	0.339	1.485	0.511	20.481	6.751
GT	21.681	87.729	N/A	N/A	N/A
DSEBMs	14.293	46.933	25.194	41.083	33.852
DAGMM	21.066	71.632	41.211	83.551	60.720
RSRAE	6.305	33.853	10.940	32.061	18.869

2.8 Related theory for the RSR penalty

We explain here why we find it natural to incorporate RSR within a neural network. In Section 2.8.1 we first review the mathematical idea of an autoencoder and discuss the robustness of a linear autoencoder with an $\ell_{2,1}$ loss (i.e., RSR loss). We then explain why a general autoencoder with an $\ell_{2,1}$ loss is not expected to be robust to outliers and why an RSR layer can improve its robustness. In Section 2.8.2 we relate the linear autoencoder minimization problem with a subspace problem. Section 2.8.3 is a first step of extending this view to a generative network. It establishes some robustness of WGAN with a linear generator, but the extension of an RSR layer to WGAN is left as an open problem. In Section 2.8.4 we shows the second term of the proposed RSR energy (2.4) may be reduced in theory. In Section 2.8.5 we discuss further potential theory for extending the RSR layer.

2.8.1 Robustness and related properties of autoencoders

Mathematically, an autoencoder for a dataset $\{\boldsymbol{x}^{(t)}\}_{t=1}^N \subset \mathbb{R}^D$ and a latent dimension $d < D$ is composed of an encoder $\mathscr{E} : \mathbb{R}^D \to \mathbb{R}^d$ and a decoder $\mathscr{D} : \mathbb{R}^d \to \mathbb{R}^D$ that minimize the following energy function with $p = 2$:

$$\sum_{t=1}^{N} \left\| \boldsymbol{x}^{(t)} - \mathscr{D} \circ \mathscr{E}(\boldsymbol{x}^{(t)}) \right\|_2^p , \tag{2.7}$$

where $\circ$ denotes function decomposition. It is a natural nonlinear generalization of PCA [69]. Indeed, in the case of a linear autoencoder, $\mathscr{E}$ and $\mathscr{D}$ are linear maps represented by matrices $\boldsymbol{E} \in \mathbb{R}^{d\times D}$ and $\boldsymbol{D} \in \mathbb{R}^{D\times d}$, respectively, that need to minimize (among such matrices) the following loss function with $p = 2$

$$\sum_{t=1}^{N} \left\| \boldsymbol{x}^{(t)} - \boldsymbol{D}\boldsymbol{E}\boldsymbol{x}^{(t)} \right\|_2^p . \tag{2.8}$$

We explain later in Section 2.8.2 that if $(\boldsymbol{D}^\star, \boldsymbol{E}^\star)$ is a minimizer of (2.8) with $p = 2$ (among $\boldsymbol{E} \in \mathbb{R}^{d\times D}$ and $\boldsymbol{D} \in \mathbb{R}^{D\times d}$), then $\boldsymbol{D}^\star\boldsymbol{E}^\star$ is the orthoprojector on the d-dimensional PCA subspace. This means, that the latent code $\{\boldsymbol{E}^\star\boldsymbol{x}^{(t)}\}_{t=1}^N$ parametrizes the PCA subspace and an additional application of $\boldsymbol{D}^\star$ to $\{\boldsymbol{E}^\star\boldsymbol{x}^{(t)}\}_{t=1}^N$ results in the

projections of the data points $\{\boldsymbol{x}^{(t)}\}_{t=1}^{N}$ onto the PCA subspace. The recovery error for data points on this subspace is zero (as $\boldsymbol{D}^{\star}\boldsymbol{E}^{\star}$ is the identity on this subspace), and in general, this error is the Euclidean distance to the PCA subspace, $\left\|\boldsymbol{x}^{(t)} - \boldsymbol{D}^{\star}\boldsymbol{E}^{\star}\boldsymbol{x}^{(t)}\right\|_2$.

Intuitively, the idea of a general autoencoder is the same. It aims to fit a nice structure, such as a manifold, to the data, where ideally $\mathscr{D} \circ \mathscr{E}$ is a projection onto this nice structure. This idea can only be made rigorous for data approximated by simple geometric structure, e.g., by a graph of a sufficiently smooth function.

In order to extend these methods to anomaly detection, one needs to incorporate robust strategies, so that the methods can still recover the underlying structure of the inliers, and consequently assign lower recovery errors for the inliers and higher recovery errors for the outliers. For example, in the linear case, one may assume a set of inliers lying on and around a subspace and an arbitrary set of outliers (with some restriction on their fraction). PCA, and equivalently, the linear autoencoder that minimizes (2.8) with $p = 2$, is not robust to general outliers. Thus it is not expected to distinguish well between inliers and outliers in this setting. As explained later in Section 2.8.2, minimizing (2.8) with $p = 1$ gives rise to the least absolute deviations subspace. This subspace can be robust to outliers under some conditions, but these conditions are restrictive (see examples in [47]). In order to deal with more adversarial outliers, it is advised to first normalize the data to the sphere (after appropriate centering) and then estimate the least absolute deviations subspace. This procedure was theoretically justified for a general setting of adversarial outliers in [70].

As in the linear case, an autoencoder that uses the loss function in (2.7) with $p = 1$ may not be robust to adversarial outliers. Unlike the linear case, there are no simple normalizations for this case. Indeed, the normalization to the sphere can completely distort the structure of an underlying manifold and it is also hard to center in this case. Furthermore, there are some obstacles of establishing robustness for the nonlinear case even under special assumptions.

Our basic idea for a robust autoencoder is to search for a latent low-dimensional code for the inliers within a larger embedding space. The additional RSR loss focuses on parametrizing the low-dimensional subspace of the encoded inliers, while being robust to outliers. Following the above discussion, we enhance such robustness by applying a normalization similar to the one discussed above, but adapted better to the structure

of the network (see Section 2.5.2). The emphasis of the RSR layer is on appropriately encoding the inliers, where the encoding of the outliers does not matter. It is okay for the encoded outliers to lie within the subspace of the encoded inliers, as this will result in large recovery errors for the outliers. However, in general, most encoded outliers lie away from this subspace, and this is why such a mechanism is needed (otherwise, a regular autoencoder may obtain a good embedding).

2.8.2 Property of linear autoencoders

In this section we characterize the solution of (2.8) via a subspace problem. Special case solutions to this problem include both the PCA subspace and the least absolute deviations subspace.

The following proposition expresses the solution of (2.8) in terms of another minimization problem. After proving it, we clarify that the other minimization problem is related to both PCA and RSR.

Proposition 2.8.1. *Let $p \geq 1$, $d < D$, and $\{\boldsymbol{x}^{(t)}\}_{t=1}^{N} \subset \mathbb{R}^D$ be a dataset with rank at least d. If $(\boldsymbol{D}^\star, \boldsymbol{E}^\star) \in \mathbb{R}^{D\times d} \times \mathbb{R}^{d\times D}$ is a minimizer of (2.8), then*

$$\boldsymbol{D}^\star \boldsymbol{E}^\star = \boldsymbol{P}^\star \ , \tag{2.9}$$

where $\boldsymbol{P}^\star \in \mathbb{R}^{D\times D}$ is a minimizer of

$$\sum_{t=1}^{N} \left\| \boldsymbol{x}^{(t)} - \boldsymbol{P}\boldsymbol{x}^{(t)} \right\|_2^p \ , \tag{2.10}$$

among all orthoprojectors $\boldsymbol{P}$ (that is, $\boldsymbol{P} = \boldsymbol{P}^T$ and $\boldsymbol{P}^2 = \boldsymbol{P}$) of rank d.

Note that when $p = 2$, the energy function in (2.10) corresponds to PCA. More precisely, a minimizer $\boldsymbol{P}^\star$ of (2.10) (among rank d orthoprojectors) is an orthoprojector on a d-dimensional PCA subspace, equivalently, a subspace spanned by top d eigenvectors of the sample covariance (we assume for simplicity linear, and not affine, autoencoder, so the PCA subspace is linear and thus when $p = 2$ the data is centered at the origin). This minimizer is unique if and only if the d-th eigenvalue of the sample covariance is larger than the $(d + 1)$-st eigenvalue. These elementary facts are reviewed in Section

II-A of [32].

When $p = 1$, the minimizer $\boldsymbol{P}^\star$ of (2.10) (among rank d orthoprojectors) is an orthoprojector on the d-dimensional least absolute deviations subspace. This subspace is reviewed in Section II-D of [32] as a common approach for RSR. The minimizer is often not unique, where sufficient and necessary conditions for local minima of (2.10) are studied in [47].

2.8.3 Relationship of the RSR loss with linearly generated WGAN

An open problem is whether RSR can be used within other neural network structures for unsupervised learning, such as variational autoencoders (VAEs) [71] and generative adversarial networks (GANs) [54]. The latter two models are used in anomaly detection with a score function similar to the reconstruction error [72, 73, 74, 75].

While we do not solve this problem, we establish a natural relationship between RSR and Wasserstein-GAN (WGAN) [55, 76] with a linear generator, which is analogous to the example of a linear autoencoder mentioned above.

Let W_p denote the p-Wasserstein distance in $\mathbb{R}^D$ ($p \geq 1$). That is, for two probability distributions μ, ν on $\mathbb{R}^D$,

$$W_p(\mu,\nu) = \left(\inf_{\pi \in \Pi(\mu,\nu)} \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim\pi} \|\boldsymbol{x}-\boldsymbol{y}\|_2^p \right)^{1/p}, \tag{2.11}$$

where $\Pi(\mu,\nu)$ is the set of joint distributions with μ, ν as marginals. We formulate the following proposition (while prove it later in Appendix A.2) and then interpret it.

Proposition 2.8.2. *Let $p \geq 1$ and $\boldsymbol{\mu}$ be a Gaussian distribution on $\mathbb{R}^D$ with mean $\boldsymbol{m}_X \in \mathbb{R}^D$ and full-rank covariance matrix $\boldsymbol{\Sigma}_X \in \mathbb{R}^{D\times D}$ (that is, $\boldsymbol{\mu}$ is $\mathcal{N}(\boldsymbol{m}_X, \boldsymbol{\Sigma}_X)$). Then*

$$\begin{aligned} \min_{\boldsymbol{\nu} \text{ is } \mathcal{N}(\mathrm{m_Y},\Sigma_\mathrm{Y})} \quad & W_p(\boldsymbol{\mu},\boldsymbol{\nu}) \\ \text{s.t.} \quad & \boldsymbol{m}_Y \in \mathbb{R}^D \\ & \mathrm{rank}(\Sigma_\mathrm{Y}) = \mathrm{d} \end{aligned} \tag{2.12}$$

is achieved when $\boldsymbol{m}_Y = \boldsymbol{m}_X$ *and* $\boldsymbol{\Sigma}_Y = \boldsymbol{P}_{\mathscr{L}} \boldsymbol{\Sigma}_X \boldsymbol{P}_{\mathscr{L}}$*, where for* $X \sim \boldsymbol{\mu}$

$$\mathscr{L} = \underset{\dim \mathscr{L} = \mathrm{d}}{\arg\min} \mathbb{E} \| X - \boldsymbol{P}_{\mathscr{L}} X \|_2^p \ . \tag{2.13}$$

The setting of this proposition implicitly assumes a linear generator of WGAN. Indeed, the linear mapping, which can be represented by a $d \times D$ matrix, maps a distribution in $\mathcal{N}(\boldsymbol{m}_X, \boldsymbol{\Sigma}_X)$ into a distribution in $\mathcal{N}(\boldsymbol{m}_Y, \boldsymbol{\Sigma}_Y)$ and reduces the rank of the covariance matrix from D to d. The proposition states that in this setting the underlying minimization is closely related to minimizing the loss function (2.3). Note that here $p \geq 1$, however, if one further corrupts the sample, then $p = 1$ is the suitable choice [32]. This choice is also more appropriate for WGAN, since there is no p-WGAN for $p \neq 1$.

Nevertheless, training a WGAN is not exactly the same as minimizing the W_1 distance [76], since it is difficult to impose the Lipschitz constraint for a neural network. Furthermore, in practice, the WGAN generator, which is a neural network, is nonlinear, and thus its output is typically non-Gaussian. The robustness of WGAN with a linear autoencoder, which we established here, does not extend to a general WGAN (this is similar to our earlier observation that the robustness of a linear autoencoder with an RSR loss does not generalize to a nonlinear autoencoder). We believe that a similar structure like the RSR layer has to be imposed for enhancing the robustness of WGAN, and possibly also other generative networks, but we leave its effective implementation as an open problem.

2.8.4 Further discussion of the RSR term

The RSR energy in (2.4) includes two different terms. The proposition below indicates that the second term of (2.4) is zero when plugging into it the solution of the minimization of the first term of (2.4) with the additional requirement that $\boldsymbol{A}$ has full rank. That is, in theory, one may only minimize the first term of (2.4) over the set of matrices $\boldsymbol{A} \in \mathbb{R}^{d \times D}$ with full rank. We then discuss computational issues of this different minimization.

Proposition 2.8.3. *Assume that* $\{\boldsymbol{z}^{(t)}\}_{t=1}^{N} \subset \mathbb{R}^D$ *spans* $\mathbb{R}^D$, $d \leqslant D$ *and let*

$$\boldsymbol{A}^\star = \underset{\substack{\boldsymbol{A}\in\mathbb{R}^{d\times D}\\ \mathrm{rank(A)=d}}}{\arg\min} \sum_{t=1}^{N} \left\| \boldsymbol{z}^{(t)} - \boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{z}^{(t)} \right\|_2. \tag{2.14}$$

Then $\boldsymbol{A}^\star \boldsymbol{A}^{\star\mathrm{T}} = \boldsymbol{I}_d$.

The minimization in (2.14) is nonconvex and intractable. Nevertheless, [50] propose a heuristic to solve it with some weak guarantees and [51] propose an algorithm with guarantees under some conditions. However, such a minimization is even more difficult when applied to the combined energy in (2.5), instead of (2.4). Therefore, we find it necessary to include the second term in (2.4) that imposes the nearness of $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}$ to an orthogonal projection (equivalently, of $\boldsymbol{A}\boldsymbol{A}^{\mathrm{T}}$ to the identity).

2.8.5 Relevant Mathematical Theory

We note that a complex network can represent a large class of functions. Consequently, for a sufficiently complex network, minimizing the loss function in (2.7) results in minimum value zero. In this case the minimizing "manifold" contains the original data, including the outliers. On the other hand, the RSR loss term imposes fitting a subspace that robustly fits only part of the data and thus cannot result in minimum value zero. Nevertheless, imposing a subspace constraint might be too restrictive, even in the latent space. A seminal work by [77] studies optimal types of curves that contain general sets. This work relates the construction and optimal properties of these curves with multiscale approximation of the underlying set by lines. It was generalized to higher dimensions in [78] and to a setting relevant to outliers in [79]. These works suggest loss functions that incorporate several linear RSR layers from different scales. Nevertheless, their pure setting does not directly apply to our setting. We have also noticed various technical difficulties when trying to directly implement these ideas to our setting.

Chapter 3

Autoencoding Mixture Posterior with Wasserstein Penalty for Novelty Detection

3.1 Introduction

In this chapter, we study a robust version of novelty detection that allows a nontrivial fraction of corrupted samples, namely outliers, within the training set. We solve this problem by using a special variational autoencoder (VAE) [80]. Our VAE is able to model the underlying distribution of the uncorrupted data, despite nontrivial corruption. We refer to our new method as "Mixture Autoencoding with Wasserstein penalty", or "MAW". In order to clarify it, we first review previous works and then explain our contributions in view of these works.

3.1.1 Previous work

Solutions to novelty detection either estimate the density of the inlier distribution [81, 82] or determine a geometric property of the inliers, such as their boundary set [64, 65, 83, 84, 85]. When the inlier distribution is nicely approximated by a low-dimensional linear subspace, [86] proposes to distinguish between inliers and outliers via Principal

Component Analysis (PCA). In order to consider more general cases of nonlinear low-dimensional structures, one may use autoencoders (or restricted Boltzmann machines), which nonlinearly generalize PCA [69][Ch. 2] and whose reconstruction error naturally provides a score for membership in the inlier class. Instances of this strategy with various architectures include [40, 39, 87, 88, 89, 90]. In all of these works, but [39], the training set is assumed to solely represent the inlier class. In fact, [88] observed that interpolation of a latent space, which was trained using digit images of a complex shape, can lead to digit representation of a simple shape. If there are also outliers (with a simple shape) among the inliers (with a complex shape), encoding the inlier distribution becomes even more difficult. Nevertheless, some previous works already explored the possibility of corrupted training set [83, 84, 39]. In particular, [83, 39] test artificial instances with at most 5% corruption of the training set and [84] considers ratios of 10%, but with very small numbers of training points. In this work we consider corruption ratios up to 33% (a fraction of 50% of outliers per inliers), with a method that tries to estimate the distribution of the training set, and not just a geometric property.

VAEs [80] have been commonly used for generating distributions with reconstruction scores and are thus natural for novelty detection without corruption. They determine the latent code of an autoencoder via variational inference [91, 92]. Alternatively, they can be viewed as autoencoders for distributions that penalize the Kullback-Leibler (KL) divergence of the latent distribution from the prior distribution. The first VAE-based method for novelty detection was suggested by [72]. It was recently extended by [93] who modified the training objective. A variety of VAE models were also proposed for special anomaly detection problems, which are different than novelty detection [94, 95, 96]. Current VAE-based methods for novelty detection do not perform well when the training data is corrupted. Indeed, the learned distribution of any such method also represents the corruption, that is, the outlier component. To the best of our knowledge, no effective solutions were proposed for collapsing the outlier mode so that the trained VAE would only represent the inlier distribution.

An adversarial autoencoder (AAE) [97] and a Wasserstein autoencoder (WAE) [98] can be considered as variants of VAE. The penalty term of AAE takes the form of a generative adversarial network (GAN) [69], where its generator is the encoder. A

Wasserstein autoencoder (WAE) [98] generalizes AAE with a framework that minimizes the Wasserstein metric between the sample distribution and the inference distribution. It reformulates the corresponding objective function so that it can be implemented in the form of an AAE.

There are two relevant lines of works on robustness to outliers in linear modeling that can be used in nonlinear settings via autoencoders or VAEs. Robust PCA aims to deal with sparse elementwise corruption of a data matrix [99, 37, 34, 33]. Robust subspace recovery (RSR) aims to address general corruption of selected data points and thus better fits the framework of outliers [42, 37, 43, 44, 45, 46, 47, 48, 49, 50, 100, 32, 70]. Autoencoders that use robust PCA for anomaly detection tasks were proposed in [41, 35]. [101] show that a VAE can be interpreted as a nonlinear robust PCA problem. Nevertheless, explicit regularization is often required to improve robustness to sparse corruption in VAEs [102, 103]. RSR was successfully applied to outlier detection by [1]. One can apply their work to the different setting of novelty detection; however, our proposed VAE formulation seems to work better.

We remark that the setting of our work is different than that of out-of-distribution (OOD) detection and open-set recognition. Indeed, in these recent settings the inliers are from multiple classes that need to be identified. On the other hand, this work does not ask to classify the inliers.

We also remark that albeit being an VAE, MAW is not designed to address the other interesting scenario: the generative task using impure training data. We plan to extend in this track in a future work.

3.1.2 This work

We propose a robust novelty detection procedure, MAW, that aims to model the distribution of the training data in the presence of nontrivial fraction of outliers. We highlight its following four features:

- MAW models the latent distribution by a Gaussian mixture of low-rank inliers and full-rank outliers, and applies the inlier distribution for testing. Previous applications of mixture models for novelty detection were designed for multiple modes of inliers and used more complicated tools such as constructing another

network [39] or applying clustering [38, 104].

- MAW applies a novel dimension reduction component, which extracts lower-dimensional features of the latent distribution. The reduced dimension allows using full covariances; whereas previous VAE-based methods for novelty detection used diagonal covariances in their models [72, 93]. The new component is inspired by the RSR layer in [1]; however, they are essentially different since the RSR layer only applies to data points and not to probability distributions.

- For the latent code penalty, MAW uses the Wasserstein-1 (W_1) metric. We prove that the Wasserstein metric gives rise to outliers-robust estimation and is suitable to the low-rank modeling of inliers by MAW. We also show that these properties do not hold for the KL divergence, which is used by VAE, AAE and WAE. To the best of our knowledge, this is the first theoretical analysis that clarifies the advantage of the Wasserstein distance over the KL divergence in a VAE in terms of robustness to outliers and low-rank inlier modeling. We remark that the use of W_1 in WAE is different than that of MAW. Indeed, in WAE W_1 measures the distance between the data distribution and the generated distribution and it does not appear in the latent code. Our use of W_1 can be viewed as a variant of AAE, which replaces GAN with Wasserstein GAN (WGAN) [55], and thus replaces the minimization of the KL divergence by that of the W_1 distance.

- MAW achieves state-of-the-art results on popular anomaly detection datasets.

Additional two features are as follows. First, for reconstruction, MAW replaces the common least squares formulation with a least absolute deviations formulation. This can be justified by the use of either a robust estimator [105] or a likelihood function with a heavier tail. Second, MAW is attractive for practitioners. It is simple to implement in any standard deep learning library, and is easily adaptable to other choices of network architecture, energy functions and similarity scores.

We remark that since we do not have labels for the training set, we cannot supervisedly learn both the inlier Gaussian component and the outlier Gaussian component. However, the use of two outliers-robust losses (least absolute deviation and the W_1 distance) allows MAW to model the inlier Gaussian component. Note that when testing, we only use this model for the inliers. In Section 3.6 we intuitively clarify, with

supporting experiments, the mechanism that helps in such modeling.

The structure of the rest of this chapter is organized as follows. We explain MAW in Section 3.2. We establish the advantage of its use of the Wasserstein metric in Section 3.3. We carefully test MAW in Section 3.4. We test the sensitivity of MAW to hyperparameters in Section 3.5. In Section 3.6 we further provide the insight of the mechanism of MAW with numerical simulations as a support.

3.2 Description of MAW

We motivate and overview the underlying model and assumptions of MAW in Section 3.2.1. We describe the implementation details of its components in Section 3.2.2. Figure 3.1 illustrates the general idea of MAW and can assist in reading this section. We summarize the algorithms of MAW for novelty detection in Section 3.2.3.

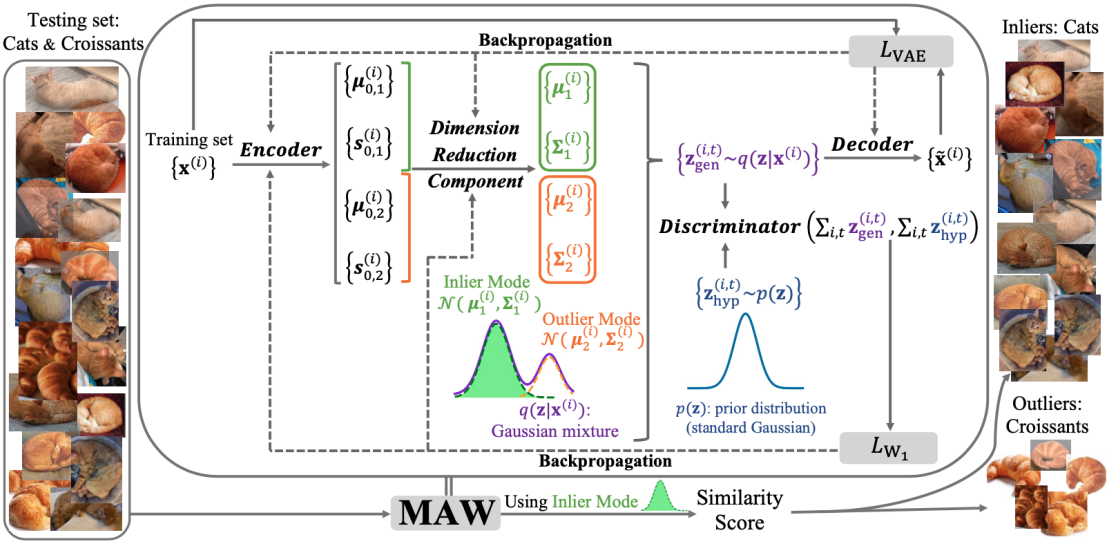


Figure 3.1: Demonstration of the architecture of MAW for novelty detection.

3.2.1 The model and assumptions of MAW

MAW aims to robustly estimate a mixture inlier-outlier distribution for the training data and then use its inlier component to detect outliers in the testing data. For this purpose, it designs a novel variational autoencoder with an underlying mixture model

and a robust loss function in the latent space. We find the variational framework natural for novelty detection. Indeed, it learns a distribution that describes the inlier training examples and generalizes to the inlier test data. Moreover, the variational formulation allows a direct modeling of a Gaussian mixture model in the latent space, unlike a standard autoencoder.

We assume L training points in $\mathbb{R}^D$, which we designate by $\{\boldsymbol{x}^{(i)}\}_{i=1}^{L}$. Let $\boldsymbol{x}$ be a random variable on $\mathbb{R}^D$ with the unknown training data distribution that we estimate by the empirical distribution of the training points. We assume a latent random variable $\boldsymbol{z}$ of low and even dimension $2 \leq d \leq D$ (our default choice is $d = 2$), and a standardized Gaussian prior, $p(\boldsymbol{z})$, so that $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_{d\times d})$. The posterior distribution $p(\boldsymbol{z}|\boldsymbol{x})$ is unknown. However, we assume an approximation to it, which we denote by $q(\boldsymbol{z}|\boldsymbol{x})$, such that $\boldsymbol{z}|\boldsymbol{x}$ is a mixture of two Gaussian distributions representing the inlier and outlier components. More specifically, $\boldsymbol{z}|\boldsymbol{x} \sim \eta\mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) + (1-\eta)\mathcal{N}(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$, where we explain next its parameters. We assume that $\eta > 0.5$, where our default value is $\eta = 5/6$, so that the first mode of $\boldsymbol{z}$ represents the inliers and the second one represents the outliers. The other parameters are generated by the encoder network and a following dimension reduction component.We remark that unlike previous works which adopted Gaussian mixtures to model the clusters of inliers [106, 39], the Gaussian mixture model in MAW aims to separate between inliers and outliers. The dimension reduction component involves a mapping from a higher-dimensional space onto the latent space. It is analogous to the RSR layer proposed by [1] that projects encoded points onto the latent space, but requires a more careful design since we consider a distribution rather than sample points. Due to this reduction, we assume that the mapped covariance matrices of $\boldsymbol{z}|\boldsymbol{x}$ are full, unlike common single-mode VAE models that assume a diagonal covariance [80, 72]. Our underlying assumption is that the inliers lie on a low-dimensional structure and we thus enforce the lower rank $d/2$ for $\boldsymbol{\Sigma}_1$, but allow $\boldsymbol{\Sigma}_2$ to have full rank d. Nevertheless, we later describe a necessary regularization of both matrices by the identity.

Following the VAE framework, we approximate the unknown posterior distribution $p(\boldsymbol{z}|\boldsymbol{x})$ within the variational family $\mathcal{Q} = \{q(\boldsymbol{z}|\boldsymbol{x})\}$, which is indexed by $\boldsymbol{\mu}_1$, $\boldsymbol{\Sigma}_1$, $\boldsymbol{\mu}_2$ and $\boldsymbol{\Sigma}_2$. A standard VAE framework would minimize the expected KL-divergence from $p(\boldsymbol{z}|\boldsymbol{x})$ to $q(\boldsymbol{z}|\boldsymbol{x})$ in $\mathcal{Q}$, where the expectation is taken over $p(\boldsymbol{x})$. By Bayes' rule this is

equivalent to maximizing the evidence lower bound (ELBO):

$$\text{ELBO}(q) = \mathbb{E}_{p(\boldsymbol{x})}\mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x})} \log p(\boldsymbol{x}|\boldsymbol{z}) - \mathbb{E}_{p(\boldsymbol{x})} KL(q(\boldsymbol{z}|\boldsymbol{x})\|p(\boldsymbol{z})) \ .$$

The first term of ELBO is the reconstruction likelihood. Its second term restricts the deviation of $q(\boldsymbol{z}|\boldsymbol{x})$ from $p(\boldsymbol{z})$ and can be viewed as a regularization term. Unlike a standard VAE, which maximizes the evidence lower bound (ELBO), MAW maximizes the following ELBO-Wasserstein, or ELBOW, function, which uses the W_1 distance:

$$\text{ELBOW}(q) = \mathbb{E}_{p(\boldsymbol{x})}\mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x})} \log p(\boldsymbol{x}|\boldsymbol{z}) - W_1(q(\boldsymbol{z}), p(\boldsymbol{z})) \ . \tag{3.1}$$

ELBOW is a more robust version of ELBO with a different regularization. That is, it replaces $\mathbb{E}_{p(\boldsymbol{x})} KL(q(\boldsymbol{z}|\boldsymbol{x})\|p(\boldsymbol{z}))$ with $W_1(q(\boldsymbol{z}), p(\boldsymbol{z}))$. We remark that the W_1 distance cannot be computed between $q(\boldsymbol{z}|\boldsymbol{x})$ and $p(\boldsymbol{z})$ and ELBOW thus practically replaces $q(\boldsymbol{z}|\boldsymbol{x})$ with its expected distribution, $q(\boldsymbol{z}) = \mathbb{E}_{p(\boldsymbol{x})} q(\boldsymbol{z}|\boldsymbol{x})$ (or a discrete approximation of this).

Following the VAE framework, we use a Monte-Carlo approximation to estimate $\mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x})} \log p(\boldsymbol{x}|\boldsymbol{z})$ with i.i.d. samples, $\{\boldsymbol{z}^{(t)}\}_{t=1}^{T}$, from $q(\boldsymbol{z}|\boldsymbol{x})$ as follows:

$$\mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x})} \log p(\boldsymbol{x}|\boldsymbol{z}) \approx \frac{1}{T} \sum_{t=1}^{T} \log p(\boldsymbol{x}|\boldsymbol{z}^{(t)}). \tag{3.2}$$

To improve the robustness of our model, we choose the negative log likelihood function $-\log p(\boldsymbol{x}|\boldsymbol{z}^{(t)})$ to be a constant multiple of the ℓ_2 norm of the difference of the random variable $\boldsymbol{x}$ and a mapping of the sample $\boldsymbol{z}^{(t)}$ from $\mathbb{R}^d$ to $\mathbb{R}^D$ by the decoder, $\mathcal{D}$, that is,

$$-\log p(\boldsymbol{x}|\boldsymbol{z}^{(t)}) \propto \left\| \boldsymbol{x} - \mathcal{D}(\boldsymbol{z}^{(t)}) \right\|_2 \ . \tag{3.3}$$

Note that we deviate from the common choice of the squared ℓ_2 norm, which corresponds to an underlying Gaussian likelihood and assume instead a likelihood with a heavier tail.

MAW trains its networks by minimizing –ELBOW(q). For any $1 \leq i \leq L$, it samples $\{\boldsymbol{z}_{\text{gen}}^{(i,t)}\}_{t=1}^{T}$ from $q(\boldsymbol{z}|\boldsymbol{x}^{(i)})$, where all samples are independent. Using the aggregation

formula:

$$q(\boldsymbol{z}) = L^{-1} \sum_{i=1}^{L} q(\boldsymbol{z}|\boldsymbol{x}^{(i)}),$$

the approximation of $p(\boldsymbol{x})$ by the empirical distribution of the training data, and (3.1)-(3.3), MAW applies the following approximation of –ELBOW(q):

$$\frac{1}{LT} \sum_{i=1}^{L} \sum_{t=1}^{T} \left\| \boldsymbol{x}^{(i)} - \mathcal{D}(\boldsymbol{z}_{\text{gen}}^{(i,t)}) \right\|_2 + W_1 \left(\frac{1}{L} \sum_{i=1}^{L} q(\boldsymbol{z}|\boldsymbol{x}^{(i)}), p(\boldsymbol{z}) \right). \tag{3.4}$$

Our procedure of minimizing (3.4) is described in Section 3.2.2. It is independent of the multiplicative constant in (3.3) and therefore this constant is ignored in (3.4).

During testing, MAW identifies outliers according to low similarity scores computed between test points and points generated from the learned inlier component of $\boldsymbol{z}|\boldsymbol{x}$.

3.2.2 Details of implementing MAW

MAW has a VAE-type structure with additional WGAN-type structure for minimizing the W_1 loss in (3.4). We provide here details of implementing these structures. Some specific choices of the networks are described in Section 3.4 since they may depend on the type of datasets.

The VAE-type structure of MAW contains three ingredients: encoder, dimension reduction component and decoder. The encoder forms a neural network $\mathcal{E}$ that maps the training sample $\boldsymbol{x}^{(i)} \in \mathbb{R}^D$ to $\boldsymbol{\mu}_{0,1}^{(i)}, \boldsymbol{\mu}_{0,2}^{(i)}, \boldsymbol{s}_{0,1}^{(i)}, \boldsymbol{s}_{0,2}^{(i)}$ in $\mathbb{R}^{D'}$, where our default choice is $D' = 128$. The dimension reduction component then computes the following statistical quantities of the Gaussian mixture $\boldsymbol{z}|\boldsymbol{x}^{(i)}$: means $\boldsymbol{\mu}_1^{(i)}$ and $\boldsymbol{\mu}_2^{(i)}$ in $\mathbb{R}^d$ and covariance matrices $\boldsymbol{\Sigma}_1^{(i)}$ and $\boldsymbol{\Sigma}_2^{(i)}$ in $\mathbb{R}^{d \times d}$. First, a linear layer, represented by $\boldsymbol{A} \in \mathbb{R}^{D' \times d}$, maps (via $\boldsymbol{A}^{\mathrm{T}}$) the features $\boldsymbol{\mu}_{0,1}^{(i)}$, $\boldsymbol{\mu}_{0,2}^{(i)} \in \mathbb{R}^{D'}$ to the following respective vectors in $\mathbb{R}^d$:

$$\boldsymbol{\mu}_1^{(i)} = \boldsymbol{A}^{\mathrm{T}} \boldsymbol{\mu}_{0,1}^{(i)} \text{ and } \boldsymbol{\mu}_2^{(i)} = \boldsymbol{A}^{\mathrm{T}} \boldsymbol{\mu}_{0,2}^{(i)}.$$

Form

$$\boldsymbol{M}_j^{(i)} = \boldsymbol{A}^{\mathrm{T}} \mathrm{diag}(\boldsymbol{s}_{0,j}^{(i)}) \boldsymbol{A} \text{ for } j = 1, 2.$$

For $j = 2$, compute

$$\boldsymbol{\Sigma}_2^{(i)} = \boldsymbol{M}_2^{(i)} \boldsymbol{M}_2^{(i)\mathrm{T}}.$$

For $j = 1$, we first need to reduce the rank of $\boldsymbol{M}_1^{(i)}$. For this purpose, we form

$$\boldsymbol{M}_1^{(i)} = \boldsymbol{U}_1^{(i)} \mathrm{diag}(\boldsymbol{\sigma}_1^{(i)}) \boldsymbol{U}_1^{(i)\mathrm{T}}, \tag{3.5}$$

the spectral decomposition of $\boldsymbol{M}_1^{(i)}$, and then truncate its bottom $d/2$ eigenvalues. That is, let $\tilde{\boldsymbol{\sigma}}_1^{(i)} \in \mathbb{R}^d$ have the same entries as the largest $d/2$ entries of $\boldsymbol{\sigma}_1^{(i)}$ and zero entries otherwise. Then, compute

$$\tilde{\boldsymbol{M}}_1^{(i)} = \boldsymbol{U}_1^{(i)} \mathrm{diag}(\tilde{\boldsymbol{\sigma}}_1^{(i)}) \boldsymbol{U}_1^{(i)\mathrm{T}} \tag{3.6}$$

and

$$\boldsymbol{\Sigma}_1^{(i)} = \tilde{\boldsymbol{M}}_1^{(i)} \tilde{\boldsymbol{M}}_1^{(i)\mathrm{T}}.$$

Since the TensorFlow package requires numerically-significant positive definiteness of covariance matrices, we add an identity matrix to both $\boldsymbol{\Sigma}_1^{(i)}$ and $\boldsymbol{\Sigma}_2^{(i)}$. Despite this, the low-rank structure of $\boldsymbol{\Sigma}_1^{(i)}$ is still evident. Note that the dimension reduction component only trains $\boldsymbol{A}$. The decoder, $\mathcal{D} : \mathbb{R}^d \to \mathbb{R}^D$, maps independent samples, $\{\boldsymbol{z}_{\mathrm{gen}}^{(i,t)}\}_{t=1}^T$, generated for each $1 \leq i \leq L$ by the distribution

$$\eta \mathcal{N}(\boldsymbol{\mu}_1^{(i)}, \boldsymbol{\Sigma}_1^{(i)}) + (1 - \eta) \mathcal{N}(\boldsymbol{\mu}_2^{(i)}, \boldsymbol{\Sigma}_2^{(i)}),$$

into the reconstructed data space.

The loss function associated with the VAE structure is the first term in (3.4). We can write it as

$$L_{\mathrm{VAE}}(\mathcal{E}, \boldsymbol{A}, \mathcal{D}) = \frac{1}{LT} \sum_{i=1}^{L} \sum_{t=1}^{T} \left\| \boldsymbol{x}^{(i)} - \mathcal{D}(\boldsymbol{z}_{\mathrm{gen}}^{(i,t)}) \right\|_2 . \tag{3.7}$$

The dependence of this loss on $\mathcal{E}$ and $\boldsymbol{A}$ is implicit, but follows from the fact that the parameters of the sampling distribution of each $\boldsymbol{z}_{\mathrm{gen}}^{(i,t)}$ were obtained by $\mathcal{E}$ and $\boldsymbol{A}$.

The WGAN-type structure seeks to minimize the second term in (3.4) using the dual formulation

$$W_1\left(\frac{1}{L}\sum_{i=1}^{L} q(\boldsymbol{z}|\boldsymbol{x}^{(i)}), p(\boldsymbol{z})\right) = \sup_{\|f\|_{\text{Lip}}\leq 1} \mathbb{E}_{\boldsymbol{z}_{\text{hyp}}\sim p(\boldsymbol{z})} f(\boldsymbol{z}_{\text{hyp}}) - \mathbb{E}_{\boldsymbol{z}_{\text{gen}}\sim \frac{1}{L}\sum_{i=1}^{L} q(\boldsymbol{z}|\boldsymbol{x}^{(i)})} f(\boldsymbol{z}_{\text{gen}}). \tag{3.8}$$

The generator of this WGAN-type structure is composed of the encoder $\mathcal{E}$ and the dimension reduction component, which we represent by $\boldsymbol{A}$. It generates the samples $\{\boldsymbol{z}_{\text{gen}}^{(i,t)}\}_{i=1,t=1}^{L,T}$ described above. The discriminator, $\mathcal{D}is$, of the WGAN-type structure plays the role of the Lipschitz function f in (3.8). It compares the latter samples with the i.i.d. samples $\{\boldsymbol{z}_{\text{hyp}}^{(i,t)}\}_{t=1}^{T}$ from the prior distribution. In order to make $\mathcal{D}is$ Lipschitz, its weights are clipped to $[-1, 1]$ during training. In the MinMax game of this WGAN-type structure, the discriminator minimizes and the generator ($\mathcal{E}$ and $\boldsymbol{A}$) maximizes

$$L_{W_1}(\mathcal{D}is) = \frac{1}{LT}\sum_{i=1}^{L}\sum_{t=1}^{T}\left(\mathcal{D}is(\boldsymbol{z}_{\text{gen}}^{(i,t)}) - \mathcal{D}is(\boldsymbol{z}_{\text{hyp}}^{(i,t)})\right) . \tag{3.9}$$

We note that maximization of (3.9) by the generator is equivalent to minimization of the loss function

$$L_{\text{GEN}}(\mathcal{E}, \boldsymbol{A}) = -\frac{1}{LT}\sum_{i=1}^{L}\sum_{t=1}^{T}\mathcal{D}is(\boldsymbol{z}_{\text{gen}}^{(i,t)}) . \tag{3.10}$$

During training, MAW alternatively minimizes the losses (3.7), (3.9) and (3.10) instead of their weighted sum. Therefore, any multiplicative constant in front of either term of (3.4) will not effect the optimization. In particular, it was okay to omit the multiplicative constant of (3.3) when deriving (3.4).

For each testing point $\boldsymbol{y}^{(j)}$, we sample $\{\boldsymbol{z}_{\text{in}}^{(j,t)}\}_{t=1}^{T}$ from the inlier mode of the learned latent Gaussian mixture and decode them as

$$\{\tilde{\boldsymbol{y}}^{(j,t)}\}_{t=1}^{T} = \{\mathcal{D}(\boldsymbol{z}_{\text{in}}^{(j,t)})\}_{t=1}^{T}.$$

Using a similarity measure $S(\cdot,\cdot)$ (our default is the cosine similarity), we compute

$$S^{(j)} = \sum_{t=1}^{T} S(\boldsymbol{y}^{(j)}, \tilde{\boldsymbol{y}}^{(j,t)}).$$

If $S^{(j)}$ is larger than a chosen threshold, then $\boldsymbol{y}^{(j)}$ is classified normal, and otherwise, novel. Additional details of MAW are in Section 3.2.3.

3.2.3 Algorithmic for MAW

Algorithms 3 and 4 describe training MAW and applying MAW for novelty detection, respectively. In these descriptions, we denote by $\boldsymbol{\theta}$, $\boldsymbol{\varphi}$ and $\boldsymbol{\delta}$ the trainable parameters of the encoder $\mathcal{E}$, decoder $\mathcal{D}$ and discriminator $\mathcal{D}is$, respectively. Recall that $\boldsymbol{A}$ includes the trained parameters of the dimension reduction component.

Algorithm 3 Training MAW

Input: Training data $\{\boldsymbol{x}^{(i)}\}_{i=1}^{L}$; initialized parameters $\boldsymbol{\theta}$, $\boldsymbol{\varphi}$ and $\boldsymbol{\delta}$ of $\mathcal{E}$, $\mathcal{D}$ and $\mathcal{D}is$, respectively; initialized $\boldsymbol{A}$; weight η; number of epochs; batch size I; sampling number T; learning rate α

Output: Trained parameters $\boldsymbol{\theta}$, $\boldsymbol{\varphi}$ and $\boldsymbol{A}$

for each epoch **do**
  for each batch $\{\boldsymbol{x}^{(i)}\}_{i\in\mathcal{I}}$ **do**
    $\boldsymbol{\mu}_{0,1}^{(i)}, \boldsymbol{\mu}_{0,2}^{(i)}, \boldsymbol{s}_{0,1}^{(i)}, \boldsymbol{s}_{0,2}^{(i)} \leftarrow \mathcal{E}(\boldsymbol{x}^{(i)})$
    $\boldsymbol{\mu}_{j}^{(i)} \leftarrow \boldsymbol{A}^{\mathrm{T}}\boldsymbol{\mu}_{0,j}^{(i)},\ \boldsymbol{M}_{j}^{(i)} \leftarrow \boldsymbol{A}^{\mathrm{T}}\mathrm{diag}(\boldsymbol{s}_{0,j}^{(i)})\boldsymbol{A},\ j=1,2$
    Compute $\tilde{\boldsymbol{M}}_{1}^{(i)}$ according to (3.5) and (3.6)
    $\boldsymbol{\Sigma}_{1}^{(i)} \leftarrow \tilde{\boldsymbol{M}}_{1}^{(i)}\tilde{\boldsymbol{M}}_{1}^{(i)\mathrm{T}},\ \boldsymbol{\Sigma}_{2}^{(i)} \leftarrow \boldsymbol{M}_{2}^{(i)}\boldsymbol{M}_{2}^{(i)\mathrm{T}}$
    for $t = 1, \cdots, T$ **do**
      sample a batch $\{\boldsymbol{z}_{\mathrm{gen}}^{(i,t)}\}_{i\in\mathcal{I}} \sim \eta\mathcal{N}(\boldsymbol{\mu}_{1}^{(i)}, \boldsymbol{\Sigma}_{1}^{(i)}) + (1-\eta)\mathcal{N}(\boldsymbol{\mu}_{2}^{(i)}, \boldsymbol{\Sigma}_{2}^{(i)})$
      sample a batch $\{\boldsymbol{z}_{\mathrm{hyp}}^{(i,t)}\}_{i\in\mathcal{I}} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$
    end for
    $(\boldsymbol{\theta}, \boldsymbol{A}, \boldsymbol{\varphi}) \leftarrow (\boldsymbol{\theta}, \boldsymbol{A}, \boldsymbol{\varphi}) - \alpha\nabla_{(\boldsymbol{\theta},\boldsymbol{A},\boldsymbol{\varphi})}L_{\mathrm{VAE}}(\boldsymbol{\theta}, \boldsymbol{A}, \boldsymbol{\varphi})$ according to (3.7)
    $\boldsymbol{\delta} \leftarrow \boldsymbol{\delta} - \alpha\nabla_{\boldsymbol{\delta}}L_{W_1}(\boldsymbol{\delta})$ according to (3.9)
    $\boldsymbol{\delta} \leftarrow \mathrm{clip}(\boldsymbol{\delta}, [-1, 1])$
    $(\boldsymbol{\theta}, \boldsymbol{A}) \leftarrow (\boldsymbol{\theta}, \boldsymbol{A}) - \alpha\nabla_{(\boldsymbol{\theta},\boldsymbol{A})}L_{\mathrm{GEN}}(\boldsymbol{\theta}, \boldsymbol{A})$ according to (3.10)
  end for
end for

Algorithm 4 Applying MAW to novelty detection

Input: Test data $\{\boldsymbol{y}^{(j)}\}_{j=1}^{N}$; sampling number T; trained MAW model; threshold $\boldsymbol{\epsilon}_{\mathrm{T}}$; similarity $S(\cdot,\cdot)$

Output: Binary labels for novelty for each $j = 1, \ldots, N$

for $j = 1, \ldots, N$ **do**
  $\boldsymbol{\mu}_{0,1}^{(j)}, \boldsymbol{s}_{0,1}^{(j)} \leftarrow \mathcal{E}(\boldsymbol{y}^{(j)})$
  $\boldsymbol{\mu}_{1}^{(j)} \leftarrow \boldsymbol{A}^{\mathrm{T}}\boldsymbol{\mu}_{0,1}^{(j)},\ \boldsymbol{M}_{1}^{(j)} \leftarrow \boldsymbol{A}^{\mathrm{T}}\mathrm{diag}(\boldsymbol{s}_{0,1}^{(j)})\boldsymbol{A}$
  Compute $\tilde{\boldsymbol{M}}_{1}^{(j)}$ according to (3.5) and (3.6)
  $\boldsymbol{\Sigma}_{1}^{(j)} \leftarrow \tilde{\boldsymbol{M}}_{1}^{(j)}\tilde{\boldsymbol{M}}_{1}^{(j)\mathrm{T}}$
  for $t = 1, \cdots, T$ **do**
    sample $\boldsymbol{z}_{\mathrm{in}}^{(j,t)} \sim \mathcal{N}(\boldsymbol{\mu}_{1}^{(j)}, \boldsymbol{\Sigma}_{1}^{(j)})$
    $\tilde{\boldsymbol{y}}^{(j,t)} \leftarrow \mathcal{D}\left(\boldsymbol{z}_{\mathrm{in}}^{(j,t)}\right)$
    compute $S(\boldsymbol{y}^{(j)}, \tilde{\boldsymbol{y}}^{(j,t)})$
  end for
  $S^{(j)} \leftarrow T^{-1}\sum_{t=1}^{T} S(\boldsymbol{y}^{(j)}, \tilde{\boldsymbol{y}}^{(j,t)})$
  if $S^{(j)} \geq \boldsymbol{\epsilon}_{\mathrm{T}}$ **then**
    $\boldsymbol{y}^{(j)}$ is a normal example
  else
    $\boldsymbol{y}^{(j)}$ is a novelty
  end if
end for
return Normality labels for $j = 1, \ldots, N$

3.3 Theoretical guarantees

We theoretically establish the superiority of using the Wasserstein distance over the KL divergence, where we leave some details (in particular proofs) to Appendix B. We formulate a mathematical setting that aims to isolate the minimization of the WGAN-type structure introduced in Section 3.2.2, while ignoring unnecessary complex components of MAW. We assume a mixture parameter $\eta > 1/2$, a separation parameter $\epsilon > 0$ and denote by $\mathcal{R}$ the regularizing function, which can be either the KL divergence or the Wasserstein distance, and by $\mathcal{S}_+^K$ and $\mathcal{S}_{++}^K$ the sets of $K \times K$ positive semidefinite and positive definite matrices, respectively. Our mathematical setting, which we motivate in Section 3.3.1, assumes $\boldsymbol{\mu}_0 \in \mathbb{R}^K$ and $\boldsymbol{\Sigma}_0 \in \mathcal{S}_{++}^K$ and requires to minimize

$$\min_{\substack{\boldsymbol{\mu}_1,\boldsymbol{\mu}_2\in\mathbb{R}^K;\boldsymbol{\Sigma}_1,\boldsymbol{\Sigma}_2\in\mathcal{S}_+^K \\ \text{s.t. } \|\boldsymbol{\mu}_1-\boldsymbol{\mu}_2\|_2\geq\epsilon}} \eta\mathcal{R}\left(\mathcal{N}(\boldsymbol{\mu}_1,\boldsymbol{\Sigma}_1),\mathcal{N}(\boldsymbol{\mu}_0,\boldsymbol{\Sigma}_0)\right) + (1-\eta)\mathcal{R}\left(\mathcal{N}(\boldsymbol{\mu}_2,\boldsymbol{\Sigma}_2),\mathcal{N}(\boldsymbol{\mu_0},\boldsymbol{\Sigma}_0)\right). \tag{3.11}$$

This minimization aims to approximate the "prior" distribution $\mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$ with a Gaussian mixture distribution. For MAW, $\boldsymbol{\mu}_0 = \mathbf{0}$ and $\boldsymbol{\Sigma}_0 = \boldsymbol{I}$, but our generalization helps clarify things. The constraint $\|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\|_2 \geq \epsilon$ distinguishes between the inlier and outlier modes and it is a realistic assumption as long as ϵ is sufficiently small.

We demonstrate a neat proposition in Section 3.3.2 which shows the Wasserstein distance is more robust than the KL divergence in the case of identical covariance matrices with full-rank. In Section 3.3.3 we present the robustness results for the case of low-rank $\boldsymbol{\Sigma}_1$. We further discuss a possible deviation of the clean theory of Proposition 3.3.2 from practice in Section 3.3.4.

3.3.1 Motivation for studying (3.11)

The implementation of any VAE or its variants, such as AAE, WAE and MAW, requires the optimization of a regularization penalty $\mathcal{R}$, which measures the discrepancy between the latent distribution and the prior distribution. This penalty is typically the KL divergence, though one may use appropriate metrics such as W_2 or W_1. Therefore, one

needs to minimize

$$\mathcal{R}\left(\frac{1}{L}\sum_{i=1}^{L} q(\boldsymbol{z}|\boldsymbol{x}^{(i)}), p(\boldsymbol{z})\right) \tag{3.12}$$

over the observed variational family $\mathcal{Q} = \{q(\boldsymbol{z}|\boldsymbol{x})\}$, which indexed by some parameters of q. Here, L is the batch size of the input data and $\sum_{i=1}^{L} q(\boldsymbol{z}|\boldsymbol{x}^{(i)})$ is its observed aggregated distribution.

Since the explicit expressions of the regularization measurements between aggregated distributions are unknown, it is not feasible to study the minimizer of (3.12). We thus consider the following approximation of (3.12):

$$\sum_{i=1}^{L} \frac{1}{L}\mathcal{R}\left(q(\boldsymbol{z}|\boldsymbol{x}^{(i)}), p(\boldsymbol{z})\right). \tag{3.13}$$

We can minimize one term of this sum at a time, that, is minimize $\mathcal{R}\left(q(\boldsymbol{z}|\boldsymbol{x}), p(\boldsymbol{z})\right)$ over $\mathcal{Q}$. This minimization strategy is common in the study of the Wasserstein barycenter problem [107, 108, 109].

One of the underlying assumptions of MAW is that the prior distribution $p(\boldsymbol{z})$ is Gaussian and $q(\boldsymbol{z}|\boldsymbol{x})$ is a Gaussian mixture. That is, $p(\boldsymbol{z}) \sim \mathcal{N}(\boldsymbol{\mu_0}, \boldsymbol{\Sigma}_0)$ and $q(\boldsymbol{z}|\boldsymbol{x}) \sim \eta\mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) + (1-\eta)\mathcal{N}(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$. This gives rise to the following minimization problem

$$\min_{\boldsymbol{\mu}_1, \boldsymbol{\mu}_2 \in \mathbb{R}^K; \boldsymbol{\Sigma}_1, \boldsymbol{\Sigma}_2 \in \mathcal{S}_+^K} \mathcal{R}\left(\eta\mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) + (1-\eta)\mathcal{N}(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2), \mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)\right). \tag{3.14}$$

Similarly to approximating (3.12) by (3.13), we approximate (3.14) by the following minimization problem:

$$\min_{\boldsymbol{\mu}_1, \boldsymbol{\mu}_2 \in \mathbb{R}^K; \boldsymbol{\Sigma}_1, \boldsymbol{\Sigma}_2 \in \mathcal{S}_+^K} \eta\mathcal{R}\left(\mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1), \mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)\right) + (1-\eta)\mathcal{R}\left(\mathcal{N}(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2), \mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)\right).$$

Recall that in MAW $\mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$ and $\mathcal{N}(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$ are associated with the inlier and outlier distribution of MAW. We further assume that there is a sufficiently small threshold $\epsilon > 0$ for which $\|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\|_2 \geq \epsilon$. This is a reasonable assumption since, in practice, if $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$ are very close, the reconstruction loss will be large. These assumptions lead to the optimization problem (3.11) proposed in Section 3.3.

3.3.2 Guarantees for (3.11) with identical covariances

Our cleanest result is when $\mathbf{\Sigma}_0$, $\mathbf{\Sigma}_1$ and $\mathbf{\Sigma}_2$ coincide. It demonstrates robustness to the outlier component by the W_1 (or W_p, $p \geq 1$) minimization and not by the KL minimization (its proof is in Section B.1).

Proposition 3.3.1. *If $\boldsymbol{\mu}_0 \in \mathbb{R}^K$, $\mathbf{\Sigma}_0 \in \mathcal{S}_{++}^K$, $\epsilon > 0$ and $1 > \eta > 1/2$, then the minimizer of (3.11) with $\mathcal{R} = W_p$, $p \geq 1$ and the additional constraint: $\mathbf{\Sigma}_0 = \mathbf{\Sigma}_1 = \mathbf{\Sigma}_2$, satisfies $\boldsymbol{\mu}_1 = \boldsymbol{\mu}_0$, and thus the recovered inlier distribution coincides with the "prior distribution". However, the minimizer of (3.11) with $\mathcal{R} = KL$ and the same constraint satisfies $\boldsymbol{\mu}_0 = \eta\boldsymbol{\mu}_1 + (1-\eta)\boldsymbol{\mu}_2$.*

That is, under the setting of Proposition 3.3.1 with $\mathcal{R} = W_1$, the estimated mean of the inlier distribution, μ_1, coincides with the mean of the prior distribution, independently of the outlier distribution. However, when $\mathcal{R} = KL$, the estimated mean of the inlier distribution is sensitive to outliers.

3.3.3 Guarantees for (3.11) with low-rank $\mathbf{\Sigma_1}$

We study the minimization problem (3.11) when $\mathbf{\Sigma}_1$ has low rank and $\mathbf{\Sigma}_2 \in \mathcal{S}_{++}^K$. We fully analyze the cases where $\mathcal{R} = W_2$ and $\mathcal{R} = KL$; however, the case where $\mathcal{R} = W_1$ is difficult to analyze and compute. We first formulate results for both cases ($\mathcal{R} = W_2$ and $\mathcal{R} = KL$), and then clarify them. When $\mathcal{R} = W_2$, we assume that the prior distribution has zero mean vector $\boldsymbol{\mu}_0 = \mathbf{0}_K \in \mathbb{R}^K$ and covariance $\mathbf{\Sigma}_0 = \boldsymbol{I}_{K\times K} \in \mathbb{R}^{K\times K}$. We further denote by $\mathbf{1}_K$ the vector $(1, \cdots, 1) \in \mathbb{R}^K$. Similarly, we denote for any $n \in \mathbb{N}$, $\mathbf{0}_n$, $\mathbf{1}_n$, $\boldsymbol{I}_{n\times n}$. For vectors $\boldsymbol{a} \in \mathbb{R}^n$ and $\boldsymbol{b} \in \mathbb{R}^m$, we denote the concatenated vector in $\mathbb{R}^{n+m}$ by $(\boldsymbol{a}; \boldsymbol{b})$.

Proposition 3.3.2. *If κ, $K \in \mathbb{N}$, $K > \kappa \geq 1$, $\epsilon > 0$, $1 > \eta > \eta^\star := \frac{K-\kappa+\epsilon^2}{K-\kappa+2\epsilon^2}$, $u^\star := \left(\frac{(K-\kappa)(1-\eta)}{\epsilon^2(2\eta-1)}\right)^{\frac{1}{3}}$, where one can note that $\eta^\star > \frac{1}{2}$ and $u^\star \in (0,1)$, then the minimizer of (3.11) with $\mathcal{R} = W_2$ and the constraints that $\mathbf{\Sigma}_1$ is of rank κ and $\mathbf{\Sigma}_2$ is of rank K, satisfies $\mathbf{0}_K = u^\star\boldsymbol{\mu_2} + (1-u^\star)\boldsymbol{\mu_1}$, $\mathbf{\Sigma}_1 = \mathrm{diag}(\mathbf{1}_\kappa; \mathbf{0}_{K-\kappa})$ and $\mathbf{\Sigma}_2 = \mathrm{diag}(\mathbf{1}_\kappa; (u^\star)^{-2}\mathbf{1}_{K-\kappa})$. Moreover, $\|\boldsymbol{\mu}_1\|_2 = u^\star\epsilon$ and $\|\boldsymbol{\mu}_2\|_2 = (1-u^\star)\epsilon$.*

Proposition 3.3.3. *If* κ, $K \in \mathbb{N}$, $K > \kappa \geq 1$, $\epsilon > 0$, $\eta > 0$, $\boldsymbol{\mu}_0$, $\boldsymbol{\mu}_1 \in \mathbb{R}^K$, $\boldsymbol{\Sigma}_0 \in \mathcal{S}_{++}^K$ *and* $\boldsymbol{\Sigma}_1 \in \mathcal{S}_+^K$, $\operatorname{rank}(\boldsymbol{\Sigma}_1) = \kappa$, *then*

$$KL(\mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)||\mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)) = \infty.$$

Therefore, the solution of (3.11) with $\mathcal{R} = KL$ *and the additional constraints* $\operatorname{rank}(\boldsymbol{\Sigma}_1) = \kappa$ *and* $\boldsymbol{\Sigma}_0 = \boldsymbol{I}$ *is ill-posed.*

Note that Proposition 3.3.2 implies that as $\eta \to 1$, $u^\star \to 0$. Hence for the inlier component $\boldsymbol{\mu}_1 \to \mathbf{0}_K$ as $\eta \to 1$ and $\boldsymbol{\Sigma}_1 = \operatorname{diag}(\mathbf{1}_\kappa; \mathbf{0}_{K-\kappa})$. Therefore, in the limit the inlier distribution has the same mean as the prior distribution. Furthermore, its covariance is obtained by an appropriate projection of the covariance $\boldsymbol{\Sigma}_0$ onto a κ-dimensional subspace, independently of η. We similarly note that as $\eta \to 1$, $\boldsymbol{\Sigma}_2 \to \operatorname{diag}(\mathbf{1}_\kappa; \boldsymbol{\infty}_{K-k})$, so that the outliers disperse.

Proposition 3.3.3 implies that the KL divergence is unsuitable for low-rank covariance modeling as it leads to an infinite value in the optimization problem.

3.3.4 Some remarks on Proposition 3.3.2

We note that the inlier and outlier covariances, $\boldsymbol{\Sigma}_1$ and $\boldsymbol{\Sigma}_2$, obtained by Proposition 3.3.2, are diagonal. Furthermore, the proof of Proposition 3.3.2 clarifies that the underlying minimization problem of this proposition may assume without loss of generality that the inlier and outlier covariances are diagonal (see e.g., (B.17), which is formulated below). On the other hand, the numerical results in Section 3.4.4 support the use of full covariances, instead of diagonal covariance. Nonetheless, we claim that the full covariances matrices of MAW comes naturally from the dimension reduction component of MAW. This component also contains trainable parameters for the covariances and they will effect the weights of the encoder, that is, will effect both the W_1 minimization and the reconstruction loss. Thus the analysis of the W_1 minimization component is not sufficient for inferring the whole behavior of MAW. For tractability purposes, the minimization in (3.11) ignores the dimension reduction component. For completeness we remark that there are two other differences between the use of (3.11) in Proposition 3.3.2 and the way it arises in MAW that may possibly also result in the advantage of using full covariance in MAW. First of all, the minimization in Proposition 3.3.2 uses

$\mathcal{R} = W_2$, whereas MAW uses $\mathcal{R} = W_1$, which we find intractable when using the rest of the setting of Proposition 3.3.2. Second of all, the optimization problem (3.11) with $\mathcal{R} = W_1$ is an approximation of the minimization of $W_1\left(\frac{1}{L}\sum_{i=1}^{L} q(\boldsymbol{z}|\boldsymbol{x}^{(i)}), p(\boldsymbol{z})\right)$ (see Section 3.3.1 for explanation), which is also intractable (even if one uses $\mathcal{R} = W_2$).

3.4 Experiments

We describe the competing methods and experimental choices in Section 3.4.1. We report on the comparison with the competing methods in Section 3.4.2. In Section 3.4.3 we report the comparison between benchmark methods when the training and testing datasets are polluted by outliers of different structures. We demonstrate the importance of the novel features of MAW in Section 3.4.4.

3.4.1 Competing methods and experimental choices

We compared MAW with the following methods (descriptions and code links are introduced below) Deep Autoencoding Gaussian Mixture Model (DAGMM) [39], Deep Structured Energy-Based Models (DSEBMs) [40], Isolation Forest (IF) [110], Local Outlier Factor (LOF) [64], One-class Novelty Detection Using GANs (OCGAN) [88], One-Class SVM (OCSVM) [111] and RSR Autoencoder (RSRAE) [1]. DAGMM, DSEBMs, OCGAN and OCSVM were proposed for novelty detection. IF, LOF and RSRAE were originally proposed for outlier detection and we thus apply their trained model for the test data.

We mention the following links (or papers with links) we used for the different codes. For DSEBMs and DAGMM we used the codes of [68]. For LOF, OCSVM and IF we used the scikit-learn [112] packages for novelty detection. For OCGAN we used its TensorFlow implementation from `https://pypi.org/project/ocgan`. For RSRAE, we adapted the code of [1] to novelty detection. All experiments were executed on a Linux machine with 64GB RAM and four GTX1080Ti GPUs.

We remark that for the neural networks based methods (DAGMM, DSEBMs, OCGAN and RSRAE), we followed similar implementation details for MAW.

For completeness, we briefly describe the benchmarks.

Deep Autoencoding Gaussian Mixture Model (DAGMM) [39] is a deep

autoencoder model. It optimizes an end-to-end structure that contains both an autoencoder and an estimator for a Gaussian mixture model. Anomalies are detected using this Gaussian mixture model. We remark that this mixture model is proposed for the inliers.

Deep Structured Energy-Based Models (DSEBMs) [40] makes decision based on an energy function which is the negative log probability that a sample follows the data distribution. The energy based model is connected to an autoencoder in order to avoid the need of complex sampling methods.

Isolation Forest (IF) [110] iteratively constructs special binary trees for the training set and identifies anomalies in the test set as the ones with short average path lengths in the trees.

Local Outlier Factor (LOF) [64] measures how isolated a data point is from its surrounding neighborhood. This measure is based on an estimation of the local density of a data point using its k nearest neighbors. In the novelty detection setting, it identifies novelties according to low density regions learned from the training data.

One-class Novelty Detection Using GANs (OCGAN) [88] is composed of four neural networks: a denoising autoencoder, two adversarial discriminators, and a classifier. It aims to adversarially push the autoencoder to learn only the inlier features.

One-Class SVM (OCSVM) [111] estimates the margin of the training set, which is used as the decision boundary for the test set. Usually it utilizes a radial basis function kernel to obtain flexibility.

Robust Subspace Recovery Autoencoder (RSRAE) [1] uses an autoencoder structure together with a linear RSR layer imposed with a penalty based on the $\ell_{2,1}$ energy. The RSR layer extracts features of inliers in the latent code while helping to reject outliers. The instances with higher reconstruction errors are viewed as outliers. RSRAE trains a model using the training data. We then apply this model for detecting novelties in the test data.

For MAW and the above four reconstruction-based methods, that is, DAGMM, DSEBMs, OCGAN and RSRAE, we use the following structure of encoders and decoders, which vary with the type of data (images or non-images). For non-images, which are mapped to feature vectors of dimension D, the encoder is a fully connected network with output channels $(32, 64, 128, 128 \times 4)$. The decoder is a fully connected

network with output channels $(128, 64, 32, D)$, followed by a normalization layer at the end. For image datasets, the encoder has three convolutional layers with output channels $(32, 64, 128)$, kernel sizes $(5 \times 5, 5 \times 5, 3 \times 3)$ and strides $(2, 2, 2)$. Its output is flattened to lie in $\mathbb{R}^{128}$ and then mapped into a 128×4 dimensional vector using a dense layer (with output channels 128×4). The decoder of image datasets first applies a dense layer from $\mathbb{R}^2$ to $\mathbb{R}^{128}$ and then three deconvolutional layers with output channels $(64, 32, 3)$, kernel sizes $(3 \times 3, 5 \times 5, 5 \times 5)$ and strides $(2, 2, 2)$.

For MAW we set the following parameters: intrinsic dimension: $d = 2$; mixture parameter: $\eta = 5/6$, sampling number: $T = 5$, and size of $\boldsymbol{A}$ (used for dimension reduction): 128×2. The matrix $\boldsymbol{A}$ and the network parameters for encoders, decoders and discriminators are initialized by the Glorot uniform initializer [113].

The neural networks within MAW are implemented with TensorFlow [114] and trained for 100 epochs with batch size 128. We apply batch normalization to each layer of any neural network. The neural networks were optimized by Adam [56] with learning rate 0.00005. For the VAE-structure of MAW, we use Adam with learning rate 0.00005.

For the WGAN-type structure discriminator of MAW, we perform RMSprop [81] with learning rate 0.0005, following the recommendation by [55] for WGAN.

For all experiments, the discriminator is a fully connected network with size (32, 64, 128, 1).

3.4.2 Comparison of MAW with state-of-the-art methods

We use six datasets for novelty detection: COVID-19 Radiography database [115], CIFAR-10 [116], Caltech101 [117], Fashion MNIST [58], KDDCUP-99 [118] and Reuters-21578 [60]. We distinguish between image datasets (COVID-19, CIFAR-10, Catlech101 and Fashion MNIST) and non-image datasets (KDDCUP-99 and Reuters-21578). Below we provide additional details on the six datasets used in our experiments.

COVID-19 (Radiography) contains chest X-ray RGB images, which are labeled according to the following three categories: COVID-19 positive, normal and bacterial Pneumonia cases. We resize the images to size 64×64 and rescale the pixel intensities to lie in $[-1, 1]$. It is publicly available in `https://www.kaggle.com/tawsifurrahman/covid19-radiography-database`.

CIFAR-10 contains 10 categories of RGB images of transportation vehicles and animals. Each image is of size 32×32 and we rescale the pixel intensities to lie in $[0, 1]$. It is publicly available in `https://www.cs.toronto.edu/~kriz/cifar.html`.

Caltech101 contains RGB images of objects from 101 categories with identifying labels. Following [1] we use the largest 11 classes and preprocess their images to have size 32×32 and rescale the pixel intensities to lie in $[-1, 1]$. It is publicly available in `http://www.vision.caltech.edu/Image_Datasets/Caltech101/`.

Fashion MNIST is an image dataset containing 10 categories of grayscale images of clothing and accessories items. Each image is of size 28×28 and we rescale the pixel intensities to lie in $[-1, 1]$. We obtained the Fashion MNIST dataset from the Keras dataset library `https://keras.io/api/datasets/fashion_mnist/`.

KDDCUP-99 is a classic dataset for intrusion detection. It contains feature vectors of connections between internet protocols and a binary label for each feature vector identifying normal vs. abnormal ones. The abnormal ones are associated with an "attack" or "intrusion". The dataset is publicly available in `http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html`.

Reuters-21578 contains 21,578 documents with 90 text categories having multi-labels. Following [1], we consider the five largest classes with single labels. We utilize the scikit-learn packages: TFIDF and Hashing Vectorizer [63] to preprocess the documents into 26,147 dimensional vectors. It is publicly available in `https://archive.ics.uci.edu/ml/datasets/reuters-21578+text+categorization+collection`.

Each dataset contains several clusters (3 for COVID-19, 10 for CIFAR-10, 11 largest ones for Caltech101, 10 for Fashion MNIST, 2 for KDDCUP-99 and 5 largest ones for Reuters-21578,). We arbitrarily fix a class and uniformly sample N training inliers and N_{test} testing inliers from that class. We let N = 160, 450, 100 , 300, 6000, 350 and N_{test} = 60, 150, 100 , 60, 1200, 140 for COVID-19, CIFAR-10, Caltech101, Fashion MNIST, KDDCUP-99 and Reuters-21578, respectively. We fix c in $\{0.1, 0.2, 0.3, 0.4, 0.5\}$, and uniformly sample outliers for training from the rest of the clusters, while maintaining a fraction of c outliers per inliers. We also fix c_{test} in $\{0.1, 0.3, 0.5, 0.7, 0.9\}$ and uniformly sample outliers from the rest of the clusters for testing, while maintaining a fraction of c_{test} per inliers. We summarize the number of inliers and outliers per dataset (for both training and testing) in Table 3.1.

Table 3.1: Numbers of inliers and outliers for training and testing used in the six datasets.

Datasets	Training		Testing	
	#Inliers (N)	#Outliers ($N \times c$)	#Inliers (N_{test})	#Outliers ($N_{\text{test}} \times c_{\text{test}}$)
COVID-19 (Radiography)	160	160 $\times c$	60	60 $\times c_{\text{test}}$
CIFAR-10	450	450 $\times c$	150	150 $\times c_{\text{test}}$
Caltech101	100	100 $\times c$	100	100 $\times c_{\text{test}}$
Fashion MNIST	300	300 $\times c$	60	60 $\times c_{\text{test}}$
KDDCUP-99	6000	6000 $\times c$	1200	1200 $\times c_{\text{test}}$
Reuters-21578	350	350 $\times c$	140	140 $\times c_{\text{test}}$

Using all possible thresholds for the finite datasets, we compute the AUC (area under curve) and AP (average precision) scores (see Appendix D for their details), while considering the outliers as "positive". For each fixed $c = 0.1, 0.2, 0.3, 0.4, 0.5$ we average these results over the values of c_{test}, the different choices of inlier clusters (among all possible clusters), and three runs with different random initializations for each of these choices. We also compute the corresponding standard deviations. We report these results in Figures 3.2 and 3.3 and further specify numerical values in Appendix E.1. We observe state-of-the-art performance of MAW in all of these datasets. In Reuters-21578, DSEBMs performs slightly better than MAW and OCSVM has comparable performance. However, these two methods are not competitive in the rest of the datasets.

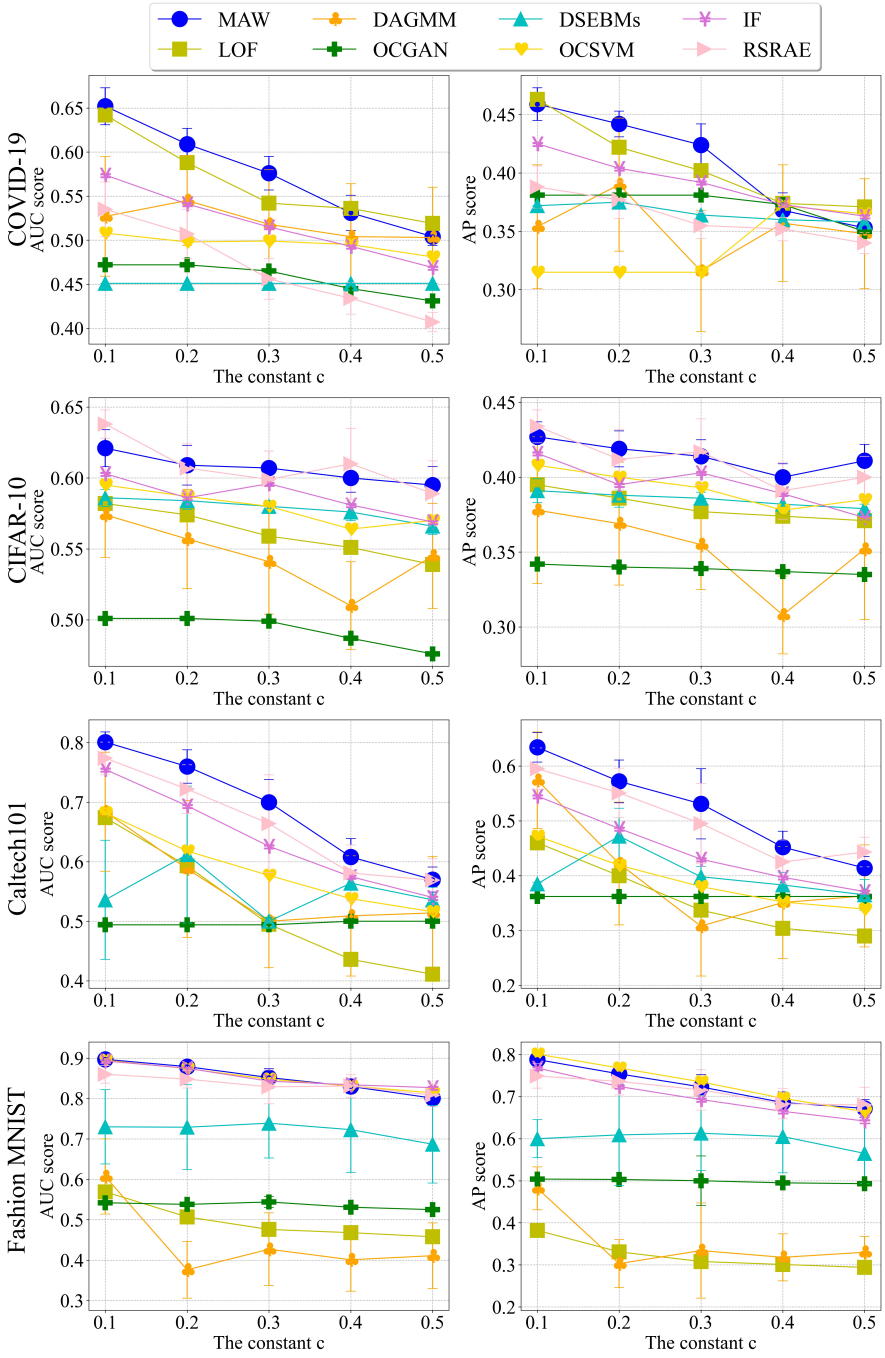


Figure 3.2: AUC (on left) and AP (on right) scores with training ratio of outliers per inliers $c = 0.1$, 0.2, 0.3, 0.4 and 0.5 for the image datasets: COVID-19, CIFAR-10, Caltech101 and Fashion MNIST.

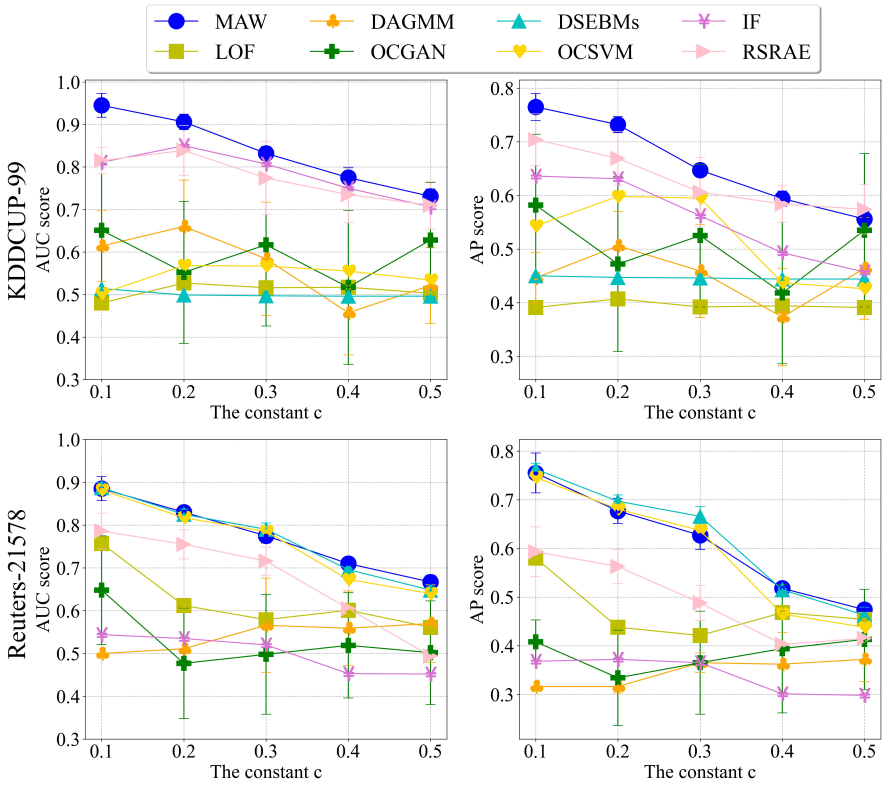


Figure 3.3: AUC (on left) and AP (on right) scores with training ratio of outliers per inliers $c = 0.1$, 0.2, 0.3, 0.4 and 0.5 for the two non-image datasets: KDDCUP-99 and Reuters-21578.

3.4.3 Experiments with different outlier types

In this section, we test the performance of MAW and the benchmark methods when the training and test sets are corrupted by outliers with different structures.

We generate a dataset, which we call "Mix Caltech101", in the following way. We fix the largest class of Caltech101 (containing airplane images) as the inlier class and randomly split it into the training inlier class (68.75 %) and testing inlier class (31.25 %). We form the training set by corrupting the training inlier class with random samples from the ten classes of CIFAR-10 [116] with training ratio of outliers per inliers $c \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$. For the test set, we corrupt the testing inlier class by "tile images" from MVTech dataset [119] with testing ratio of outliers per inliers c_{test} in $\{0.1, 0.3, 0.5, 0.7, 0.9\}$. The rest of the settings of the experiments are identical to the description in Section 3.4.2. We present the AUC and AP scores and their standard deviations in Figure 3.4.

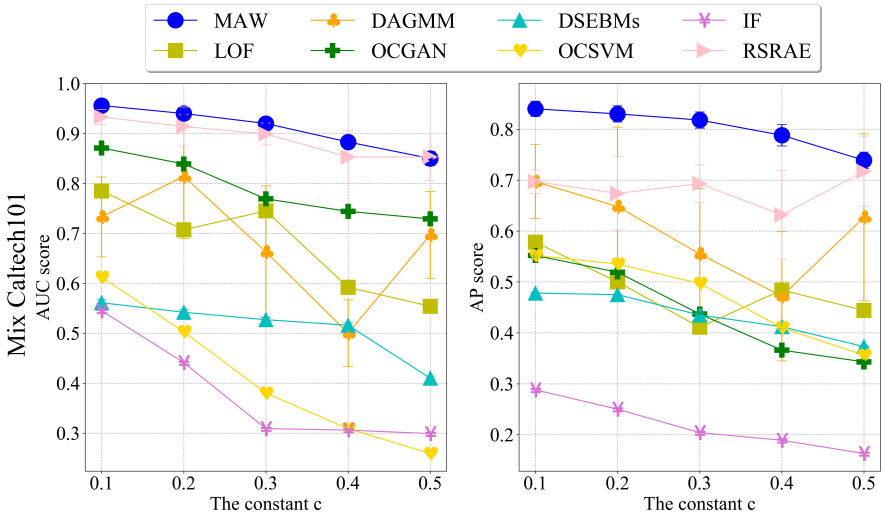


Figure 3.4: AUC and AP scores with training ratio of outliers per inliers $c \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$ for the Mix Caltech101 dataset.

The competitive advantage of MAW in comparison to the rest of the methods is also noticeable in this setting. We note that OCSVM, the traditional distance-based method, and IF, the traditional density-based method, perform poorly in this scenario, whereas they performed well in our original setting.

3.4.4 Testing the effect of the novel features of MAW

We experimentally validate the effect of the following five new features of MAW: the least absolute deviation for reconstruction, the W_1 metric for the regularization of the latent distribution, the Gaussian mixture model assumption, full covariance matrices resulting from dimension reduction component and the lower rank constraint for the inlier mode. The following methods respectively replace each of the above component of MAW with a traditional one: MAW-MSE, MAW-KL divergence, MAW-same rank, MAW-single Gaussian and MAW-diagonal cov., respectively. In addition, we consider a standard variational autoencoder (VAE). We provide additional details on each of these variants of MAW.

- **MAW-MSE** replaces the least absolute deviation loss L_{VAE} with the common mean squared error (MSE). That is, it replaces $\left\|\boldsymbol{x}^{(i)} - \mathcal{D}(\boldsymbol{z}_{\text{gen}}^{(i,t)})\right\|_2$ in (3.7) with $\left\|\boldsymbol{x}^{(i)} - \mathcal{D}(\boldsymbol{z}_{\text{gen}}^{(i,t)})\right\|_2^2$.

- **MAW-KL divergence** replaces the Wasserstein distance in (3.8) with the KL-divergence. This is implemented by replacing the WGAN-type structure of the discriminator with a standard GAN.

- **MAW-same rank** uses the same rank d for both the covariance matrices $\boldsymbol{\Sigma}_1^{(i)}$ and $\boldsymbol{\Sigma}_2^{(i)}$, instead of forcing $\boldsymbol{\Sigma}_1^{(i)}$ to have lower rank $d/2$.

- **MAW-single Gaussian** replaces the Gaussian mixture model for the latent distribution with a single Gaussian distribution with a full covariance matrix.

- **MAW-diagonal cov.** replaces the full covariance matrices resulting from the dimension reduction component by diagonal covariances. Its encoder directly produces 2-dimensional means and diagonal covariances (one of rank 1 for the inlier mode and one of rank 2 for the outlier mode).

- **VAE** has the same encoder and decoder structures as MAW. Instead of a dimension reduction component, it uses a dense layer which maps the output of the encoder to a 4-dimensional vector composed of a 2-dimensional mean and 2-dimensional diagonal covariance. This is common for a traditional VAE.

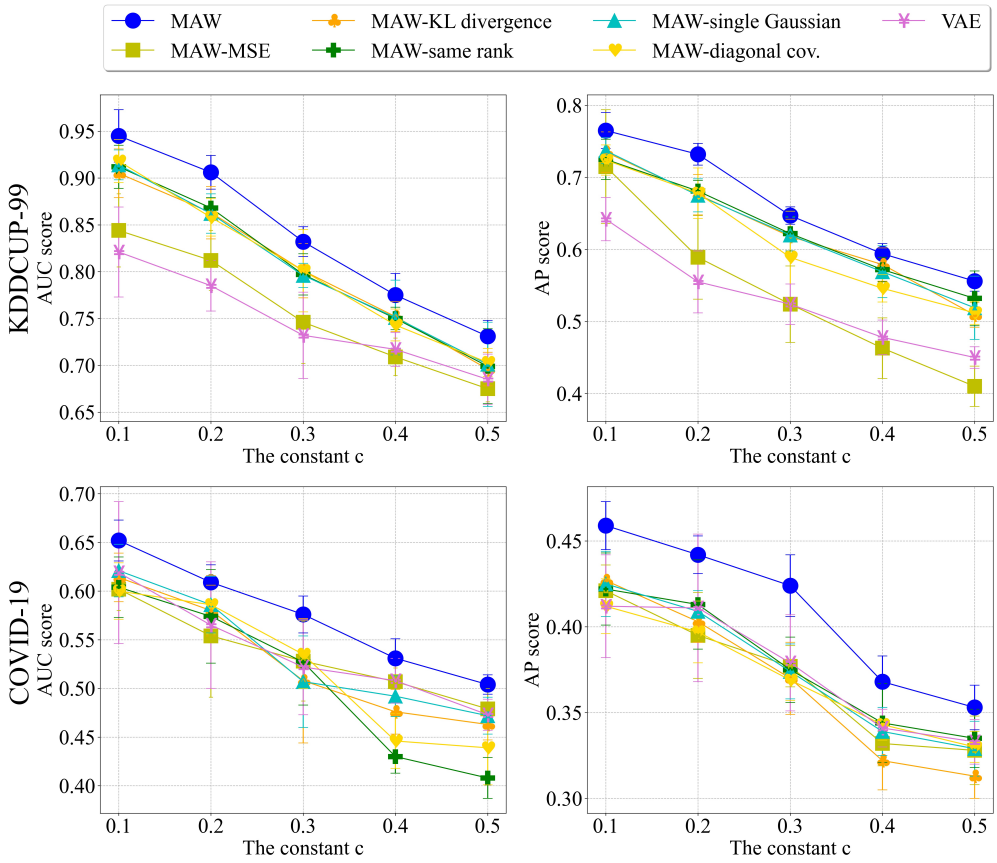


Figure 3.5: AUC (on left) and AP (on right) scores for variants of MAW (missing a novel component) with training ratio of outliers per inliers $c \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$, using KDDCUP-99 and COVID-19.

We compared the above six methods with MAW using two datasets: KDDCUP-99 and COVID-19 with training ratio of outliers per inliers $c = 0.1, 0.2, 0.3$, 0.4 and 0.5. We followed the experimental setting described in §3.4.1. Figure 3.5 reports the averages and standard deviations of the computed AUC and AP scores, where the corresponding numerical values are further recorded in Appendix E.2. The results indicate a clear decrease of accuracy when missing any of the novel components of MAW or using a standard VAE.

3.5 Sensitivity of hyperparameters

We examine the sensitivity of some of the reported results to changes of some hyperparameters. In Section 3.5.1, we report the sensitivity to choices of the intrinsic dimension. In Section 3.5.2, we report the sensitivity to choices of the mixture parameter.

3.5.1 Sensitivity to different intrinsic dimensions

In all of the other experiments in this paper the default value of the intrinsic dimension is $d = 2$. Here we study the sensitivity of our numerical results to the following choices intrinsic dimensions: $d = 2$, 4, 8, 16, 32 and 64, while using the KDDCUP-99 and COVID-19 datasets. The training ratio of outliers per inliers c are in $\{0.1, 0.2, 0.3, 0.4, 0.5\}$. We compute the AUC and AP scores averaged over the testing ratios of outliers per inliers, $c_{\text{test}} = 0.1$, 0.3, 0.5, 0.7 and 0.9, and over three runs of the same setting. Figure 3.6 reports the averaged results and their standard deviations, which are indicated by error bars.

We can see from Fig. 3.6 that lager intrinsic dimensions generally result in better performances. However, the margins of the results between different intrinsic dimensions are not large. We remark that it requires much more computation efforts for training when the dimensions are higher.

3.5.2 Sensitivity to mixture parameters

In the rest of our experiments the default value of the mixture parameter η is $5/6$. Namely, we assume that the inlier mode has larger weight among the Gaussian mixture. In this section, we study the sensitivity of the accuracy of MAW to the mixture parameters: $\{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 5/6, 0.9\}$. We use $5/6 \approx 0.83$, instead of the nearby value 0.8, since it was already tested for MAW. The training ratios of outliers per inliers are 0.1, 0.2, 0.3, 0.4 and 0.5. We report results on both KDDCUP-99 and COVID-19 in Figure 3.7.

We notice that the AUC and AP scores mildly increase as the mixture parameter η increases (though they may slightly decrease at 0.9). It seems that MAW learns well the inlier mode with a sufficiently large inlier weight, where the variation in the accuracy as a function of η is not large in general.

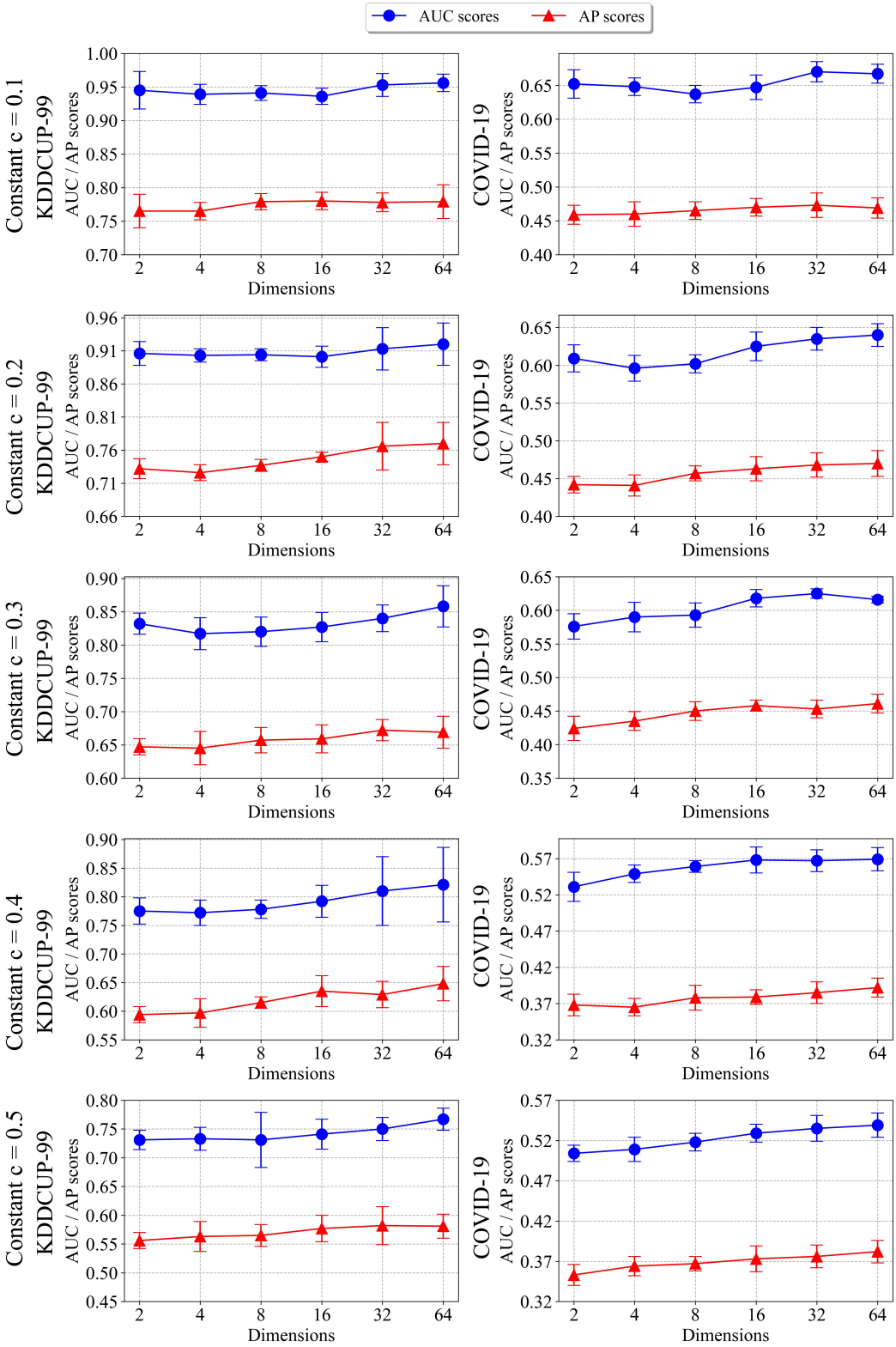


Figure 3.6: AUC and AP scores with intrinsic dimensions $d = 2$, 4, 8, 16, 32 and 64 for KDDCUP-99 (on the left) and COVID-19 (on the right), where $c \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$.

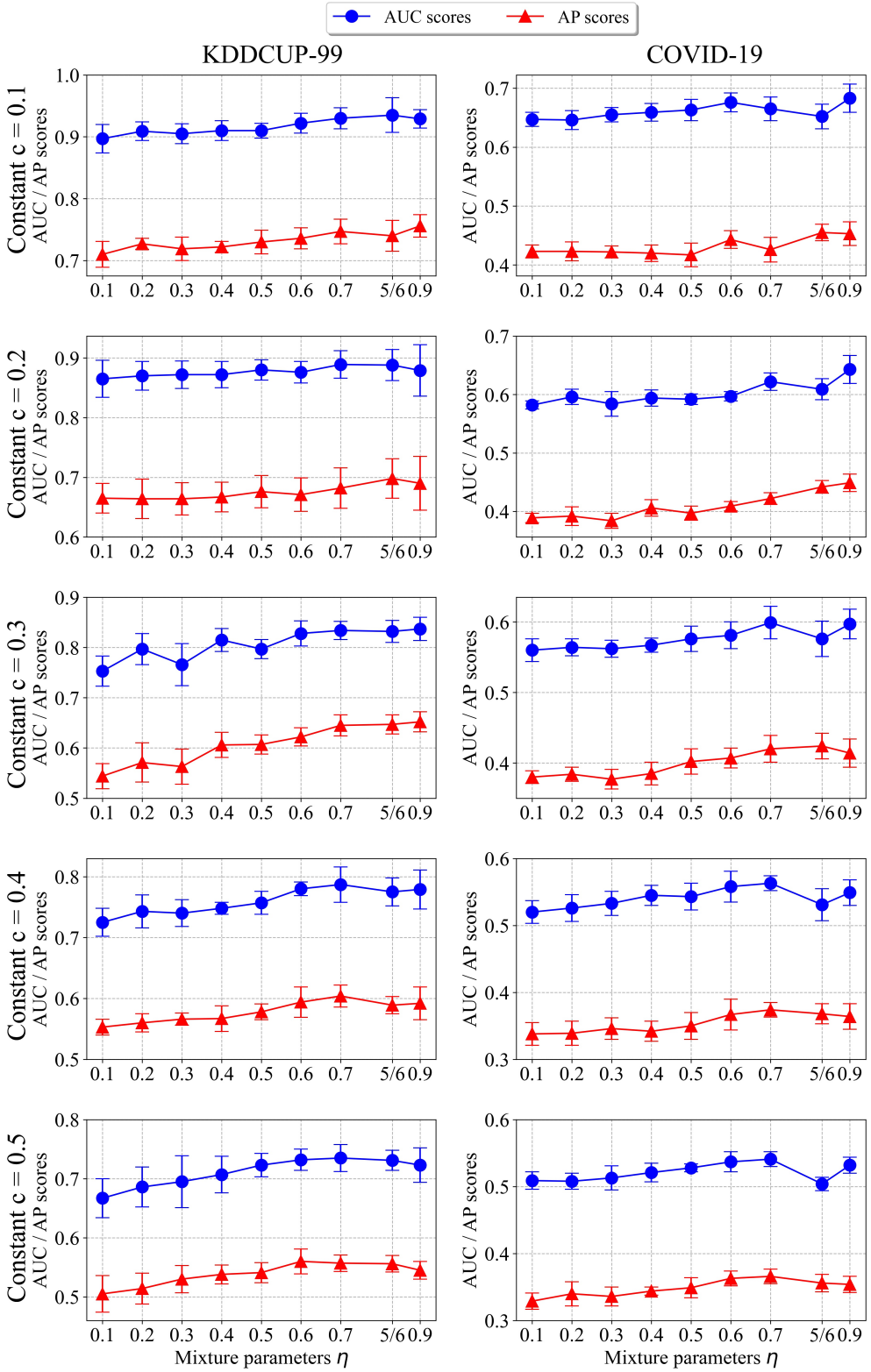


Figure 3.7: AUC and AP scores with mixture parameters $\eta = 0.1$, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 5/6 and 0.9 for KDDCUP-99 (on the left) and COVID-19 (on the right). From the top to the bottom row, the training ratios of outliers per inliers are $c = 0.1$, 0.2, 0.3, 0.4 and 0.5, respectively.

3.6 Insights on the Mechanism of MAW

We explain the basic mechanism of MAW for unsupervised alignment of the inliers with the inlier mode. Since we do not have labels for the training set, we cannot supervisedly determine the inlier mode of the latent distribution. Nevertheless, the robust losses (the least absolute deviation and the W_1 distance) guide the estimation of the inlier mode as they help in ignoring the effect of the outliers. Absolute least deviation metrics have been shown to be robust to outliers in special mathematical settings [105, 32, 1]. The robustness of the Wasserstein distance within a mathematical setting was studied in Section 3.3. Here we would like to provide some intuition how the complex procedure of MAW succeeds by using these robust metrics.

Assume that the inliers are sampled from a distribution on a low-dimensional manifold that can be encoded by a Gaussian on a low-dimensional latent space. Assume further that the outliers are arbitrary, but their percentage is smaller than that of the inliers. Given this assumption and considering the latent space, MAW aims to model the mixture component of the inliers as a Gaussian with low-rank covariance (and that of the outliers as a Gaussian with full-rank covariance). In order to provide some technical intuition for this model and show that it can fit the assumed data, let us suppose on the contrary that during training inliers and outliers are assigned to the wrong modes and show that this can either not happen or will be corrected.

We first assume a case of collapse during training, where both the inliers and outliers are modeled (in the latent space) by a Gaussian distribution with a low-rank covariance. In this case, the W_1 distance is minimized over a smaller set (due to the constraint on the rank of the outlier mode) and thus the loss is increased.

We next assume another case of collapse during training, where both the inliers and outliers are modeled (in the latent space) by a full-rank Gaussian. In this case it is most likely that the minimizer for the inliers will be full-rank, and thus due to the assumed low-dimensional structure of the inliers, it will result in an increase of the reconstruction error.

At last, assume that during training the inliers are modeled (in the latent space) by a Gaussian with full-rank covariance and the outliers are modeled (in the latent space) by a Gaussian with a low-rank covariance. One can note that this will increase the

reconstruction loss.

To further support our claim that the Gaussian mixture model is helpful for separating inliers and outliers in the latent space, we investigate the reconstruction errors of two different models. The first is MAW and the second is a variant of MAW, which replaces the Gaussian mixture model for the latent distribution with a single Gaussian distribution, whose covariance matrix has full rank. We refer to the latter model as MAW-single Gaussian. We use the KDDCUP-99 dataset with 1,000 inliers and 300 outliers in the training set, where the intial training of MAW (or MAW-single Gaussian) is the same as in Section 3.4 of the main manuscript. In Figure 3.8, we demonstrate the reconstruction error distribution of data points according to the following five scenarios.

1. **MAW, inliers and inlier distribution** (in blue) **:** Apply the trained MAW (with the corrupted model) to the inliers of the training set, while using only the inlier mode in the latent code and compute the reconstruction error between the output and the input (the ℓ_2 norm of their difference).

2. **MAW, inliers and outlier distribution** (in orange) **:** Follow the same steps as above, but replace the inlier mode with the outlier mode.

3. **MAW, outliers and inlier distribution** (in green) **:** Follow the same steps of the first case, but replace the inliers (input of the trained MAW) with the outliers.

4. **MAW-single Gaussian and inliers** (in pink) **:** Follow the same steps of the first case, but replace MAW with MAW-single Gaussian.

5. **MAW-single Gaussian and outliers** (in light purple) **:** Follow the same steps as in the above method, but replace the inliers (as input of the trained MAW-single Gaussian) with the outliers.

We can see from cases 1 and 2 above (which appear on the left of Figure 3.8) that if we try to reconstruct the inliers, then the reconstruction errors with the outlier mode are higher than those with the inlier mode. In particular, it is obvious that the inlier and outlier modes are different and do not collapse. Although we did not supervisedly train the inlier and outlier modes, it seems that the inliers align well with the inlier distribution. Moreover, comparing cases 1 and 3 above (still left of

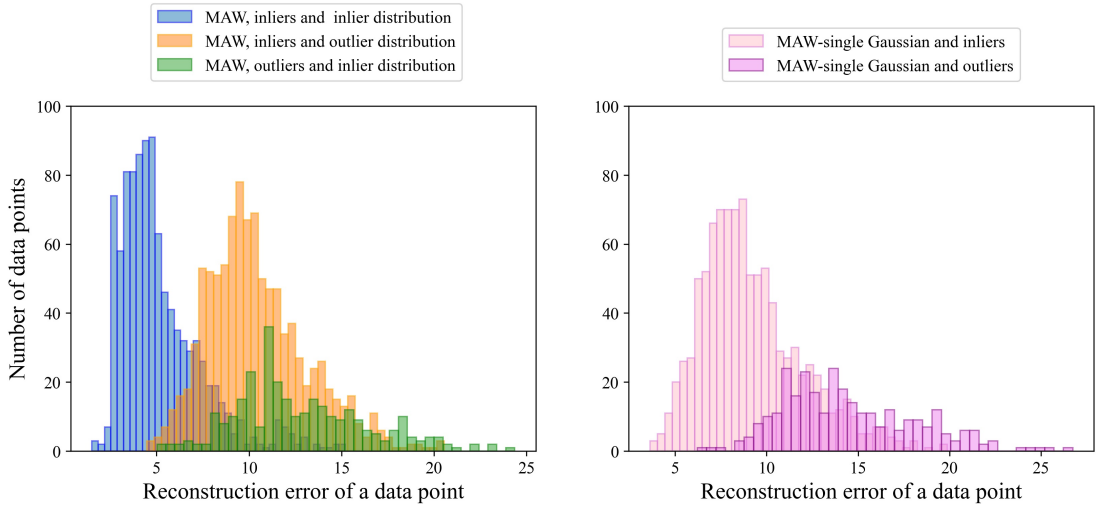


Figure 3.8: Demonstration of the distributions of the three types of reconstruction errors obtained with MAW (left) and the two types of reconstruction errors obtained with MAW-single Gaussian (right).

Figure 3.8), we can nicely distinguish between the distributions of the reconstruction errors of the inliers and the outliers. On the other hand, cases 4 and 5 (on the right of Figure 3.8) indicate that when using MAW-single Gaussian instead of MAW, the distributions of reconstruction errors of the inliers and outliers are indistinguishable. This experiment thus demonstrates the effectiveness of the Gaussian mixture model of MAW in separating the inliers and outliers for this particular experiment.

Chapter 4

Unlocking Inverse Problems Using Deep Learning: Breaking Symmetries in Phase Retrieval

4.1 Introduction

In this chapter, We will propose a novel methodology for preprocessing the training dataset to improve the end-to-end deep learning approaches of solving the phase retrieval problem. We bring out in Section 1.2 that the intrinsic symmetries of the systems might be problematic for the end-to-end approach to solve the inverse problems. We first introduce the setting of the problem in a high level viewpoint and later we will focus on the specific phase retrieval problems.

Let $\mathscr{X}$ and $\mathscr{Y}$ be two metric spaces. Let f be a continuous mapping from $\mathscr{X}$ to $\mathscr{Y}$. Given $y \in \mathscr{Y}$ which serves as the observed output. A traditional way to formulate the inverse problem determined by the mapping f is as the following optimization problem

$$\min_{x \in \mathscr{X}} \ \ell(y, f(x)) + \lambda \Omega(x) \tag{4.1}$$

where ℓ is any measurement defined on $\mathscr{Y}$ (for example, ℓ could be the mean square error if $\mathscr{Y}$ is a norm space), λ is a hyper-parameter and Ω is a regularization function.

With the assistance of deep learning, a modern data-driven end-to-end way of solving

the inverse problems is by sampling a sufficiently large amount of x_i's to form a dataset $\{(x_i, f(x_i))\}$ and implementing this sample set to train neural networks to approximate the "inverse relation" from $\mathscr{Y}$ to $\mathscr{X}$ [120, 121, 122].

However, when f is not invertible, there might be multiple estimated inputs x's in $\mathscr{X}$ which are mapped to a given observation $y \in \mathscr{Y}$. That is, the fiber $f^{-1}(\{y\})$ is not a singleton. In this scenario, the training process of the neural networks may be unstable and may further lead to unsatisfactory input estimations (we further illustrate this phenomena with simulations in Section 4.5).

We focus on a common type of the one-to-many property of the "inverse relation" that the forward mapping f admits intrinsic symmetries. That is, we suppose that there is a group $\mathcal{G}$ acting on $\mathscr{X}$ (with action $*$) so that

1. f is $\mathcal{G}-$invariant: $f(x) = f(g * x)$ for any $x \in \mathscr{X}$ and $g \in \mathcal{G}$;
2. every fiber $f^{-1}(\{y\})$ is a homogeneous $\mathcal{G}-$space: for any elements x and $\tilde{x}$ in $f^{-1}(\{y\})$, there is a $g \in \mathcal{G}$ so that $x = g * \tilde{x}$.

To address the instability of training raised by the one-to-many inverse relation, our general idea is to search for an "appropriate" subset $\mathscr{R} \subset \mathscr{X}$ where it is composed of a single "ideal" representative point from each fiber. We then aim to develop a simple mechanism to preprocess the samples x_i in $\mathscr{X}$ onto $\mathscr{R}$. By doing this, we expect the following property hold:

A. (continuous retrieval) $\mathscr{R}$ is connected and there is a continuous function h that maps from $\mathscr{Y}$ onto $\mathscr{R}$ which is the inverse of $f\big|_{\mathscr{R}}$;

We remark that we can rephrase the definition of $\mathscr{R}$ as the following two properties

B. (representative) for any $x \in \mathscr{X}$, one can find a representative element $\omega \in \mathscr{R}$ and $g \in \mathcal{G}$ such that $g * \omega = x$.

C. (smallestness) for any $\omega \in \mathscr{R}$ and any $\tilde{\omega} \in \mathscr{R} \setminus \{\omega\}$, there is no $g \in \mathcal{G}$ so that $g * \tilde{\omega} = \omega$ holds.

The first property ideally will avoid the oscillatory approximation to the inverse relation. The second and third property reduce the training samples to a minimal subset $\mathscr{R}$ according to symmetries but it still maintains requested information.

We refer the above procedure to as *symmetry breaking*. For a general mapping f and a symmetry system given by $\mathcal{G}$, it is difficult to identify a representative set $\mathscr{R}$ and not to say a systematic way to preprocess points in $\mathscr{X}$ onto $\mathscr{R}$. As a initial work, we focus on phase retrieval problems, a specific type of inverse problems, in this chapter. Further extensions, for example systems of inverse problems with individual or mutual symmetry systems and robust inverse retrieval to the observation noises, will be considered as a future work and is further discussed in Section 5.

4.1.1 A simple example of symmetry breaking

To elaborate our idea further, we look at an easy example which we would like to train a DNN to take square root. We randomly draw sufficiently many $x \in \mathbb{R}$ and thereby generate training samples $\left\{\left(x_i, x_i^2\right)\right\}$. We use these samples to train the DNN and one might hope ultimately the trained DNN output a good estimate of the square root for any input, up to sign. However, it turns out the DNN might not be stable during the training process, and hence, it might not be able to produce a descent approximation of the square-root function. This is due to a simple observation that for two points x_i^2 and x_j^2 that are near, the corresponding x_i and x_j may be close in magnitude but differ in sign. This implies that the function determined by the training data points is highly oscillatory (see Figure 4.1) and behaves like a function with many points of discontinuity—DNNs with continuous or even smooth activation functions will struggle when approximating these irregular functions. We further note that the more train samples one gathers, the more serious the problem is. A naive attempt to solve the problem is considering the positive ray $\mathbb{R}_+ \subset \mathbb{R}$. The set $\mathbb{R}_+$ is obviously connected and we observe that any $x \in \mathbb{R}$ (except 0) can be represented by an element in $\mathbb{R}_+$ by a sign flipping and it cannot be made smaller to remain representative by sign flipping operation. The training data set $\left\{x_i, x_i^2\right\}$ then will be processed so that any pair $\left(x_i, x_i^2\right)$ with negative x_i will be changed to $\left\{-x_i, x_i^2\right\}$, with the rest data points unchanged. Obviously after this processing to the training set, the function determined will closely trace the upper branch of the square root function, which is much more smooth.

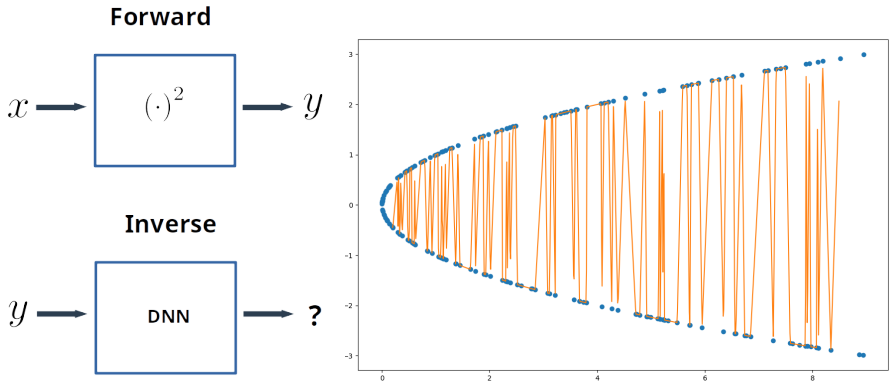


Figure 4.1: Learn to take square root. (Left) The forward and inverse models; (Right) The function (in orange) determined by the training points.

4.1.2 Our contribution

The square-root example is of course contrived and an easy fix for the problem is we only take positive (or negative) x_i's. For general inverse problems with symmetries, so long as the symmetries can relate remote inputs to the same output, for example, all the symmetries we discussed in the three quick examples in Section 1.2, the above issue of approximating highly irregular functions arises. It is a natural question if our easy fix for learning square root can be generalized. In this work,

- We take the generalized phase retrieval problem as an example, and show that effective symmetry breaking can be performed for both the real-valued and complex-valued versions of the problem. We also corroborate our theory with extensive numerical experiments.

- By working out the example, we identify the basic principle of effective symmetry breaking, which can be applied to other inverse problems with symmetries.

- We then focus on the end-to-end approach applied to nonlinear inverse problems, and concentrate on the Fourier phase retrieval (FPR) problem —which is central to scientific imaging [123] and it will be introduced in Section 4.2.

The structure of this chapter is organized as the followings. In Section 4.2 we describe the 2D FPR problem that we will mainly focus on in this chapter. In Section 4.3 we

elaborate our idea for the square-root example further and demonstrate it via two simplified phase retrieval problems. In Section 4.4 we propose our methodology, symmetry breaking, to the 2D FPR problem which releases the pain of irregular approximation of the inverse operation. It is also supported by a mathematical setting. In Section 4.5, we test our proposed methodology and show its superiority to other benchmark methods.

4.2 2D Fourier phase retrieval problem

In this section, we will introduce the 2D FPR problem.

Definition 4.2.1. *A Fourier matrix of size $m \times n$, denoted as $\boldsymbol{F}_{m\times n}$, is a matrix mapping from $\mathbb{C}^n$ to $\mathbb{C}^m$ so that for $1 \le k_1 \le m$ and $1 \le k_2 \le n$, its (k_1, k_2)-th entry is given by*

$$\boldsymbol{F}_{m\times n}(k_1, k_2) := e^{-i\frac{2\pi}{m}k_1k_2}.$$

A Fourier matrix $\boldsymbol{F}_{m\times n}$ is called oversampling if $m \ge 2n$.

We consider the forward mapping f defined as $f(\cdot) := \left|\boldsymbol{F}_{m_1\times n_1}(\,\cdot\,)\boldsymbol{F}^{\mathrm{T}}_{m_2\times n_2}\right|^2 : \mathbb{C}^{n_1\times n_2} \to \mathbb{R}^{m_1\times m_2}$. Given any observed magnitude $\boldsymbol{Y} \in \mathbb{R}^{m_1\times m_2}_{\ge 0}$, a 2D FPR problem aims at recovering a signal $\boldsymbol{X} = \{\boldsymbol{X}(k_1, k_2)\}_{\substack{k_1=0,1\cdots,n_1-1\\ k_2=0,1,\cdots,n_2-1}} \in \mathbb{C}^{n_1\times n_2}$ so that it satisfies

$$\boldsymbol{Y} = \left|\boldsymbol{F}_{m_1\times n_1}\boldsymbol{X}\boldsymbol{F}^{\mathrm{T}}_{m_2\times n_2}\right|^2, \tag{4.2}$$

where $\boldsymbol{F}_{m_1\times n_1}$ and $\boldsymbol{F}_{m_2\times n_2}$ are the Fourier matrices. We observe that applying any composition of the following operations to $\boldsymbol{X}$ will leave the observation $\boldsymbol{Y}$ unchanged.

1 2D translation of $\boldsymbol{X}$: $\boldsymbol{X}(k_1+l_1, k_2+l_2)$ for $k_1 = 0, 1\cdots, n_1-1$ and $k_2 = 0, 1, \cdots, n_2-1$. By moduling out n_1 and n_2 for the two components, the indices remain being less than n_1 and n_2, respectively;

2 2D conjugate flipping of $\boldsymbol{X}$: $\overline{\boldsymbol{X}(-n_1, -n_2)}$;

3 global phase transfer to $\boldsymbol{X}$: $\boldsymbol{X}e^{i\theta}$ for any $\theta \in [0, 2\pi)$.

Figure 4.2 illustrates the first two symmetries, assuming $\boldsymbol{X}$ is a real-valued image.

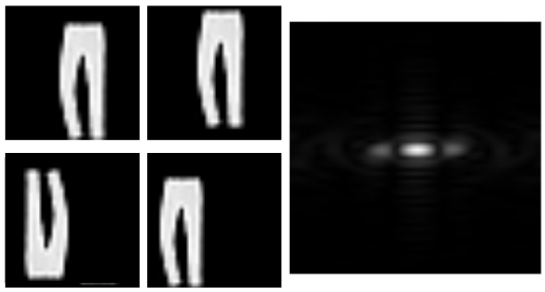


Figure 4.2: Symmetries in 2D PR. (Left) shifted and flipped copies of the same image; (Right) their common Fourier magnitude

4.3 Examples – real and complex Gaussian phase retrieval

Instead of directly solving the FPR (4.2), we first demonstrate our idea for solving it via two simplified PR problems in this section. We consider a special forward mapping $f(\cdot) := |\boldsymbol{A}(\cdot)|^2$, where $\boldsymbol{A}$ is either a real or complex i.i.d. Gaussian (with each of its entry i.i.d. sampled from Gaussian) of arbitrary means and covariances. The absolute-square operator $|\cdot|^2$ is applied elementwise of the vector. We then have the following two simplified phase retrieval (PR) problems depend on $\boldsymbol{A}$ is real or complex matrix.

Real Gaussian PR The forward model:

$$\boldsymbol{y} = |\boldsymbol{A}\boldsymbol{x}|^2 \quad \text{for real i.i.d. Gaussian } \boldsymbol{A} \in \mathbb{R}^{m\times n}, \tag{4.3}$$

where $\boldsymbol{x} \in \mathbb{R}^n$ and $\boldsymbol{y} \in \mathbb{R}^m$. The only symmetry is sign, as $\boldsymbol{x}$ and $-\boldsymbol{x}$ are mapped to the same $\boldsymbol{y}$.

Complex Gaussian PR The forward model: The forward model:

$$\boldsymbol{y} = |\boldsymbol{A}\boldsymbol{x}|^2 \quad \text{for complex i.i.d. Gaussian } \boldsymbol{A} \in \mathbb{C}^{m\times n}, \tag{4.4}$$

where $\boldsymbol{x} \in \mathbb{C}^n$ and $\boldsymbol{y} \in \mathbb{R}^m$. The only symmetry is global phase transfer, as $e^{i\theta}\boldsymbol{x}$ for all $\theta \in [0, 2\pi)$ are mapped to the same $\boldsymbol{y}$.

We remark that these two versions have been intensively studied in the recent developments of generalized PR in [124, 125, 126].

4.3.1 Real Gaussian phase retrieval

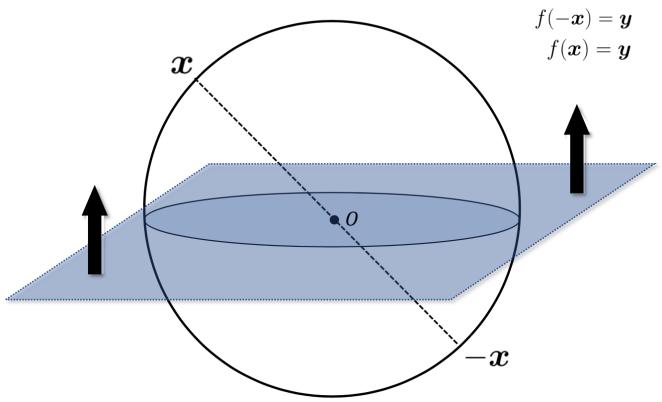


Figure 4.3: Symmetry breaking for real Gaussian phase retrieval.

In this section we focus on the real Gaussian case (4.3). Recall that in our learning square root example, the sign ambiguity caused the irregularity in the function determined by the training samples. A similar problem occurs here. For two samples that are close in the observation, say $\boldsymbol{y}$ and $\boldsymbol{y}+\boldsymbol{\delta}$ for a small $\boldsymbol{\delta}$, the signals of interest may be $\boldsymbol{x}$ and $-(\boldsymbol{x}+\boldsymbol{\delta}')$ for a small $\boldsymbol{\delta}'$. Thus, for the function our DNN tries to approximate, a small perturbation $\boldsymbol{\delta}$ in the variable leads to $2\boldsymbol{x}-\boldsymbol{\delta}'$ change to the function value, and sharp changes of this kind happen frequently as we have many training samples.

We generalize our solution of the square-root example to the real Gaussian PR. We recall that in the square-root example, the symmetry is the sign flipping and we naively break it by restricting the range of desired DNN output to $\mathbb{R}_+$. In the real Gaussian PR, the only symmetry is the global sign flipping of vectors and we observe that any pair of antipodal points map to the same observation. Thus, an intuitive generalization is to make a hyperplane cut and preprocess training samples to reside only on one side of the hyperplane which is illustrated in Figure 4.3.

In $\mathbb{R}^3$, we can see directly from Figure 4.3 that the upper half space cut out by the xy-plane is connected. Moreover, it is representative as any point in the space (except for the plane itself) can be represented by a point in this set by appropriate global sign adjustment, and it cannot be made smaller to remain representative. We formulate

the following simple proposition which says that these properties also hold for high-dimensional spaces. The proof is straightforward but is shown in Appendix C for the completeness.

Proposition 4.3.1. *Let*

$$\mathscr{R}^\star = \{\boldsymbol{x} = (x_1, \cdots, x_n) \in \mathbb{R}^n : x_n > 0\}, \tag{4.5}$$

$$Z = \{\boldsymbol{x} = (x_1, \cdots, x_n) \in \mathbb{R}^n : x_n = 0\}. \tag{4.6}$$

Then the following properties hold:

(i) ***(connected)*** *$\mathscr{R}^\star$ is connected in $\mathbb{R}^n$;*

(ii) ***(representative)*** *Z is of measure zero and for any $\boldsymbol{x} \in \mathbb{R}^n \setminus Z$, either $\boldsymbol{x} \in \mathscr{R}^\star$ or $-\boldsymbol{x} \in \mathscr{R}^\star$. That is, $\mathscr{R}^\star$ can represent any point except for those on Z by flipping sign;*

(iii) ***(smallest)*** *If we remove any single point $\boldsymbol{x}$ of $\mathscr{R}^\star$, then there is no other points in $\mathscr{R}^\star \setminus \{\boldsymbol{x}\}$ that can represent $\boldsymbol{x}$. Namely, such resulted set is not "representative" anymore.*

The coordinate hyperplane Z we use is arbitrary, and we can prove similar results for any hyperplane passing through the origin. The set Z is negligible, as the probability of sampling a point exactly from Z is zero. In fact, we can break the symmetry in Z also by recursively applying the current idea. For the sake of simplicity and in view of the probable diminishing return, we will not pursue the refined scheme here.

We explain at a heuristic level to see if the proposed method will improve things. Imagine that we have collected a set of training samples $\{\boldsymbol{x}_i, |\boldsymbol{A}\boldsymbol{x}_i|^2\}$ for real Gaussian PR. Now we are going to preprocess the data samples according to the above hyperplane cut: for all $\boldsymbol{x}_i$'s, if $\boldsymbol{x}_i$ lies above Z, we simply leave it untouched; if $\boldsymbol{x}_i$ lies below Z, we switch the sign of $\boldsymbol{x}_i$; if $\boldsymbol{x}_i$ happens to lie on Z, we make a small perturbation to $\boldsymbol{x}_i$ and then adjusts the sign as before accordingly. Now $\boldsymbol{x}_i \in R$ for all i. Since R is a connected set, when there are sufficiently dense training samples, small perturbations to $|\boldsymbol{A}\boldsymbol{x}|^2$ always only lead to small perturbations to $\boldsymbol{x}_i$. So we now have a nicely behaved target function to approximate using a DNN. Also, $\mathscr{R}^\star$ being representative implies that a sufficiently dense sample set should enable reasonable learning.

The set of three properties is also necessary for effective symmetry breaking and learning. Being representative is easy to understand. If the representative set is not the smallest, symmetry is still present for certain points in the set and so symmetry breaking is not complete. Now the set can be smallest representative but not connected. An example in the setting of proposition 4.3.1 would be taking out a small strict subset of $\mathscr{R}^\star$, say $B \subsetneq \mathscr{R}^\star$, and consider the set $M := (-B) \cup (\mathscr{R}^\star \setminus B)$. It is easy to verify that M is smallest representative, but not connected. This leaves us the trouble of approximating (locally) highly oscillatory functions.

4.3.2 Complex Gaussian phase retrieval

We now move to the complex case and deal with a different kind of symmetry. Recall that in the complex Gaussian PR, $\mathrm{e}^{i\theta}\boldsymbol{x}$ for all $\theta \in [0, 2\pi)$ are mapped to the same $|\boldsymbol{A}\boldsymbol{x}|^2$, i.e., global phase transfer is the symmetry. These "equivalent" points form a continuous curve in the complex space, contrasting the isolated antipodal point pairs in the real case.

We rephrase the desired properties for symmetry breaking in this context.

Definition 4.3.1 (representative)**.** *Fix a target subset $\mathcal{T}$ of $\mathbb{C}^n$. Let $\mathscr{R}$ be a subset of T and $\mathcal{G}$ be a group acting on $\mathbb{C}^n$ with the action $*$. We say that $\mathscr{R}$ is a representative subset for $\mathcal{T}$ if the following holds: there is a measure zero subset Z of $\mathcal{T}$ such that for any $\boldsymbol{x} \in \mathcal{T} \setminus Z$, there is a $g \in \mathcal{G}$ and an $\boldsymbol{x}' \in \mathscr{R}$ such that $\boldsymbol{x} = g * \boldsymbol{x}'$.*

In particular, the global phase transfer operation of the complex Gaussian PR is equivalent to acting the group $\mathcal{G} := \left\{\mathrm{e}^{i\theta} : \theta \in [0, 2\pi)\right\}$ on $\mathcal{T} := \mathbb{C}^n$ with complex multiplication as its action. In this context, a subset $\mathscr{R}$ is representative if except for a negligible subset of $\mathbb{C}^n$, any element of $\mathbb{C}^n$ can be represented by an element of $\mathscr{R}$ after appropriate global phase transfer.

Definition 4.3.2 (smallest representative)**.** *Fix a target subset $\mathcal{T}$ of $\mathbb{C}^n$ and let $\mathscr{R}$ be a subset of $\mathbb{C}^n$. We say that $\mathscr{R}$ is a smallest representative subset for $\mathcal{T}$ if it is representative and no element in $\mathscr{R}$ can be represented by a distinct element of $\mathscr{R}$.*

To construct a smallest representative set for $\mathcal{T} := \mathbb{C}^n$, it is helpful to start with low dimensions. When $n = 1$, any ray stemming from the origin (with origin removed)

is a smallest representative subset for $\mathbb{C}$. For simplicity, we can take the positive axis $\mathbb{R}_+$. When $n = 2$, it is natural to use the building block $\mathbb{R}_+$ for $\mathbb{C}$ and start to consider product constructions of the form $\mathbb{R}_+ \times B \subset \mathbb{C}^2$ with $B \subset \mathbb{C}$. Similarly for high dimensions, we try constructions of the form $\mathbb{R}_+ \times B \subset \mathbb{C}^n$ with $B \subset \mathbb{C}^{n-1}$. Another consideration is the measure-zero set. In the real case, we used a coordinate hyperplane. Here, as a natural generalization, we take a complex hyperplane:

$$Z = \{\boldsymbol{x} = (x_1, \cdots, x_n) \in \mathbb{C}^n : x_1 = 0\}. \tag{4.7}$$

The question now is how to choose B to make $\mathbb{R}_+ \times B$ a smallest representative subset for $\mathbb{C}^n \setminus Z$.

It turns out we actually do not get many choices. The following result says that real positivity assumed for the first coordinate constrains the construction significantly and the rest of coordinates are forced to be the entire complex space $\mathbb{C}^{n-1}$. Its proof can be found in Appendix C.2.

Proposition 4.3.2. *If $\mathscr{R}^\star := \mathbb{R}_+ \times B$ with $B \subset \mathbb{C}^{n-1}$ is a representative subset for $\mathbb{C}^n \setminus Z$, then $B = \mathbb{C}^{n-1}$.*

We now focus on this candidate set

$$\mathscr{R}^\star := \{\boldsymbol{x} = (x_1, \cdots, x_n) \in \mathbb{C}^n : \mathrm{Im}(x_1) = 0, x_1 > 0\}. \tag{4.8}$$

Our next proposition confirms that this is indeed a good choice. The proof is presented in Appendix C.3

Proposition 4.3.3. *The set $\mathscr{R}^\star$ defined in (4.8) is a connected, smallest representative set for $\mathbb{C}^n$ with Z defined as in (4.7) and identified as the negligible subset in Definition 4.3.1. We notet that Z is a measure-zero subset of $\mathbb{C}^n$.*

So our construction $\mathscr{R}^\star$ enjoys the three desired properties, similar to the real case, despite that the problem symmetry is different here. Once we emulate the data preprocessing step for the real case, i.e., all $\boldsymbol{x}_i$'s for the training data points $\{(\boldsymbol{x}_i, |\boldsymbol{A}\boldsymbol{x}_i|^2)\}$ are mapped into $\mathscr{R}^\star$, we obtain an effective symmetry breaking algorithm for complex Gaussian PR.

4.4 Breaking symmetries for FPR

FPR (4.2) has three symmetries as discussed in Section 4.2. Under the global phase transfer, equivalent data points form continuous curves that are easy to represent algebraically. The conjugate 2D flipping and nonzero content translation, however, induce irregular equivalent sets that are hard to represent. Following [4] and prescribing a rule for symmetry breaking in the original input space of $\boldsymbol{X}$ seems hopeless.

Fortunately, the three symmetries can be equivalently represented in the complex phase $\mathrm{e}^{i\theta}$ space after the Fourier transform. Let $\mathcal{X}$ denote the oversampled Fourier transform of $\boldsymbol{X}$. In the "phase" space, we have corresponding operations as in Section 4.2 which make the given observation $\boldsymbol{Y}$ unchanged. We denote the group action by $*$, where the group $\mathcal{G}$ is generated by $g_{\text{trans.}}$, $g_{\text{flip.}}$ and $g_{\text{flip.}}$ with their definitions specified below.

(1') 2D translation $g_{\text{trans.}}$ induces

$$g_{\text{trans.}} * \mathcal{X}(k_1, k_2) = e^{i2\pi\left(\frac{k_1 l_1}{m_1} + \frac{k_2 l_2}{m_2}\right)} \mathcal{X}(k_1, k_2)$$

for any $l_1, l_2 \in \mathbb{Z}$ (any allowable 2D translation of indices);

(2') 2D conjugate flipping $g_{\text{flip.}}$ induces

$$g_{\text{flip.}} * \mathcal{X} = \overline{\mathcal{X}}.$$

In terms of the complex phase, $g_{\text{flip.}}$ leads to

$$g_{\text{flip.}} * e^{i\boldsymbol{\Theta}} = \mathrm{e}^{-i\boldsymbol{\Theta}};$$

(3') Global phase transfer $g_{\text{phase.}}$ induces

$$g_{\text{phase.}} * \mathcal{X} = \mathrm{e}^{i\theta}\mathcal{X}.$$

We note that the effect of operations (2') and (3') are relatively simple: the change due to (2') is a global sign flipping in the phase space and the equivalent set due to (3')

is a line in the phase space. However, (1') is still relatively irregular whether represented in the angle or phase space.

Our strategy here is a combination of "rigorous" symmetry breaking for (2') and (3') in the complex phase space and heuristic symmetry breaking for (1') in the original space — our later real-data experiments confirm that the combination is effective. To break (1'), we propose to heuristically simply center the nonzero content. To break (2') and (3'), we perform a geometric construction of a connected, smallest representative subset in the angle space and then represent it in the phase space to avoid the tricky 2π periodicity issue in the angle space.

Consider the following set in the phase domain

$$\mathscr{R}^\star := \left\{ \boldsymbol{\Phi} \in \mathbb{C}^{m_1 \times m_2} \ : \ \boldsymbol{\Phi}(1,1) = 1, \ \boldsymbol{\Phi}(1,2) \in \mathbb{S}_+, \boldsymbol{\Phi}(i,j) \in \mathbb{S} \quad \text{for any other index } (i,j) \right\},$$

where $\mathbb{S}$ denotes the unit circle in the complex plane $\mathbb{C}$ and $\mathbb{S}_+$ the upper half circle. We can prove the following, stated in the equivalent vector space for convenience. We write $\mathcal{R}^\star \subset \mathbb{S}^{m_1 m_2}$ to mean the linear-isomorphism copy of $\mathscr{R}^\star$ as a set of vectors in $\mathbb{C}^{m_1 m_2}$. We formulate the following proposition and its proof is in Appendix C.4.

Proposition 4.4.1. *Consider the conjugate flipping and global phase transfer symmetries only. The set $\mathcal{R}^\star$ is a connected, smallest representative in the phase domain $\mathbb{S}^{m_1 m_2}$ with a negligible set $\mathcal{N} = \{1\} \times \{\Phi \in \mathbb{S} : \mathrm{Im}(\Phi) = 0\}^{m_1 m_2 - 1}$.*

To apply this, we work with end-to-end DNNs that directly predicts the $m_1 \times m_2$ complex phases. We first center the nonzero content inside $\boldsymbol{X}_i$'s in the training set, and then take the oversampled Fourier transform and perform the symmetry breaking as implied by Proposition 4.4.1 in the complex phase space. For any phase matrix $\boldsymbol{\Phi}$, the symmetry breaking goes naturally as follows: first a global phase transfer is performed to make $\Phi(1,1) = 1$, and then a global angle (here we assume the angle has been transferred to the range of $[-\pi, \pi)$) negation is performed, i.e., $\boldsymbol{\Phi} \mapsto -\boldsymbol{\Phi}$ if the second angle is negative.

We remark that for general inverse problems, although the symmetries might be very different than here and the sample spaces could also be more complicated, the three properties, which concern only the geometric and topological aspects of the space, can be generalized as a basic mathematical principle for effective symmetry breaking.

Our symmetry-breaking solution for PR also suggests that for problems with multiple symmetries, one may need to look at a transformed space, or even mixture of spaces for different symmetries for efficient representation and symmetry breaking.

4.5 Numerical experiments

Table 4.1: Test error (MSE) using different symmetry schemes

	U-Net-B	U-Net-A (ours)
No Symmetry	**0.103**	**0.103**
Flipping Symmetry	0.168	**0.162**
Shift Symmetry	0.249	**0.102**
Shift & Flipping Symmetry	0.248	**0.161**

In this section, we set up a preliminary experiment to verify our claim that effective symmetry breaking facilitates efficient learning. Particularly, we show that symmetry breaking substantially improves PR performance over alternative methods.

We conduct our experiments on the Fashion MNIST dataset [58]. We take their $60,000$ training images and $10,000$ test images to construct our training and test sets respectively. Each example is a 28×28 grayscale image. To simulate the typical black ground that causes the translation freedom in PR applications, we place all the images in a black background of 42×42 — most previous methods overlook this in their experiments, but practically the translation freedom, or what PR community call support estimation, is a major failing factor for most PR methods. So $n = 42$, and we take $m = 96$ here to ensure injectivity of the forward model $2n - 1 = 83$ is exceeded. We create 4 variants of the dataset to test the impact of symmetries on learning — this is the first time this kind of rigorous evaluation is performed. Most previous methods use natural image datasets where the image contents are naturally centered and oriented, which does not match the scenarios in PR applications, for example, in coherent diffraction imaging. We do this by modifying the images as described below, followed by the standard operation of taking squared Fourier magnitudes.

- **No Symmetry**: all images are placed in the center of the black background. Namely, we pad 7 pixels on all side of images. Samples are shown in Figure 4.4 (a)-left;

- **Flipping symmetry**: all images are placed in the center of the black background and 50% of randomly selected training and test images are 2D flipped. Samples are shown in Figure 4.4 (b)-left.
- **Shift symmetry**: all images are placed in a larger dark background and randomly translated. Samples are shown in Figure 4.4 (c)-left;
- **Shift and Flipping symmetries**: we randomly flip the images and it is followed by random translation; Samples are shown in Figure 4.4 (d)-left.

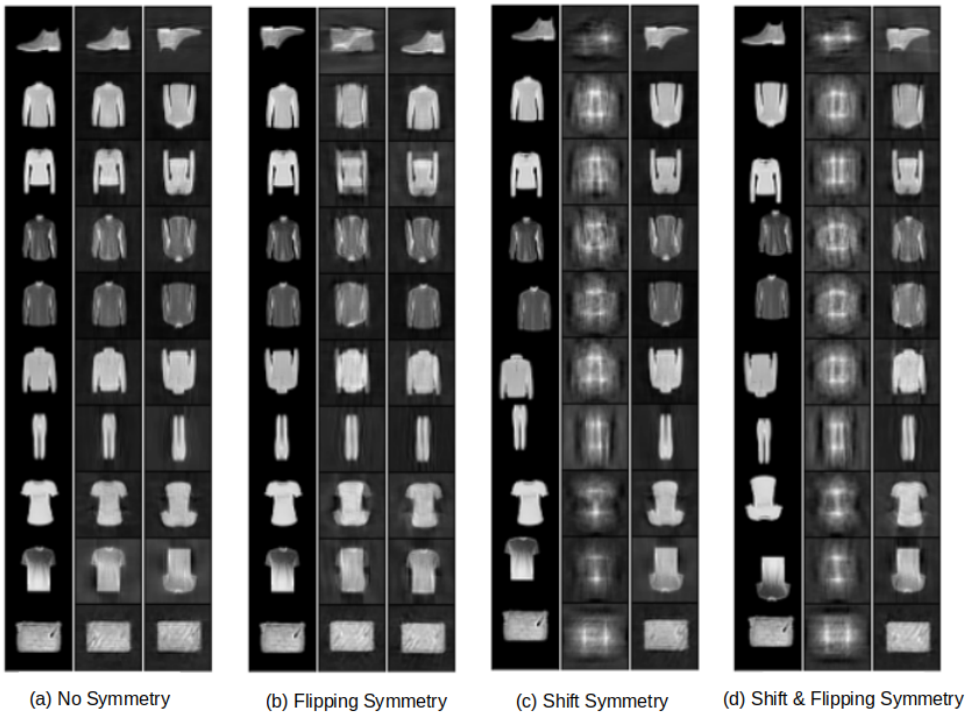


Figure 4.4: Visualization of recovery results of four different cases.

We present the results on randomly selected test images in Figure 4.4. We use U-Net [127] as our backbone neural network. We refer to the method used in [128], one of the state-of-the-art methods based on the end-to-end approach without symmetry breaking as **U-Net-$\boldsymbol{B}$** (The B means before applying symmetry breaking). We refer to our method which is with symmetry breaking as **U-Net-$\boldsymbol{A}$** (The A means after implementing symmetry breaking).

For each variant of the dataset, the left column is the groundtruth image, and the middle and right columns are reconstructions produced by U-Net-B and U-Net-A, respectively.

First note in Figure 4.4 that when no explicit symmetries are built into the dataset, U-Net-B, a representative end-to-end method for PR [128], gives good recovery. But it fails once we build in the essential symmetries. The mode of failure is interesting, as the estimated images are almost always the superposition of the symmetric (translated or flipped) copies of the groundtruth. This is very similar to the failure mode of the classic iterative methods on PR. Moreover, for images that are visually similar between the original and the flipped copy such as "handbag", "leggings", the reconstruction results are good with or without the flipping symmetry, consistent with our intuition.

On the other hand, irrespective of the symmetries, U-Net-A consistently leads to good recovery. Table 4.1 provides the average MSE adjusted to the symmetries (defined in D.2) for the testing set. Those cases with superior performances are marked as bold. As noted above, absent symmetries, both U-Net-B and U-Net-A work well and the average MSEs are the same. However, once the dataset contains the essential symmetries, we see a substantial gap in the MSEs of the reconstructed images, which is consistent with the visual results.

Method	MSE
ALM	0.299
U-Net-B	0.249
U-Net-A	**0.160**

Table 4.2: Comparison of MSE errors between our method U-Net-A and benchmark methods ALM and U-Net-B.

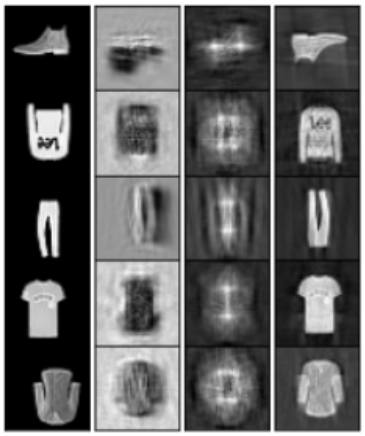


Figure 4.5: Comparison between groundtruth and reconstructed images via ALM, U-Net-B and U-Net-A, (from left to right) respectively.

For practical PR, mostly iterative methods are deployed. However, these methods are known to fail when there is translation freedom in the image and the support (for example, location of nonzero pixels) of the image content is not precisely known. To see if our end-to-end approach makes progress on this, we compare it with a state-of-the-art iterative method for PR recently proposed in [129] that has demonstrated good numerical stability and competitive performance, dubbed ALM. Here we only experiment with the most realistic version of the dataset, i.e., with both shift and flipping symmetries. Results on randomly selected test images are presented in Figure 4.5 and quantitatively results are presented in Table 4.2 (the best performance method is marked as bold). Visually, our method faithfully reconstructs the holistic content of the original images, whereas both U-Net-B and ALM fail miserably. Quantitatively, our method leads the other two by a considerable gap in MSE.

These results show that symmetry breaking is significant in unlocking the true potential of the end-to-end approach for solving PR in particular. We expect our strategy is extendable to general nonlinear inverse problems with symmetries but we leave this as a future work.

Chapter 5

Conclusion and Discussion

In this chapter, we briefly summarize the works in Chapter 2, 3 and 4, respectively. Moreover, we provide potential extensions to them.

In Chapter 2, we constructed a simple but effective RSR layer within the autoencoder structure for anomaly detection. It is easy to use and adapt. We have demonstrated competitive results for image and document data and believe that it can be useful in many other applications.

There are several directions for further exploration of the RSR loss in unsupervised deep learning models for anomaly detection. First, we are interested in theoretical guarantees for RSRAE. A more direct subproblem is understanding the geometric structure of the "manifold" learned by RSRAE. Second, it is possible that there are better geometric methods to robustly embed the manifold of inliers. For example, one may consider a multiscale incorporation of RSR layers, which we expand on in Section 2.8.5. Third, one may try to incorporate an RSR layer in other neural networks for anomaly detection that use nonlinear dimension reduction. We hope that some of these methods may be easier to directly analyze than our proposed method. For example, we are curious about successful incorporation of robust metrics for GANs or WGANs. In particular, we wonder about extensions of the theory proposed here for WGAN when considering a more general setting.

In Chapter 3, we introduced MAW, a robust VAE-type framework for novelty detection that can tolerate high corruption of the training data. We proved that the

Wasserstein distance used in MAW has better robustness to outliers and is more suitable to a low-dimensional inlier component than the KL divergence. We demonstrated state-of-the-art performance of MAW with a variety of datasets and experimentally validated that omitting any of the new ideas results in a significant decrease of accuracy.

We hope to further extend our proposal in the following ways. First of all, we plan to extend and test some of our ideas for the different problem of robust generation, in particular, for building generative networks which are robust against adversarial training data. Second of all, we would like to carefully study the virtue of our idea of modeling the most significant mode in a training data. In particular, when extending the work to generation, one has to verify that this idea does not lead to mode collapse. Furthermore, we would like to explore any tradeoff of this idea, as well as our setting of robust novelty detection, with fairness. At last, we hope to further extend our theoretical guarantees. For example, two problems that currently seem intractable are the study of the W_1 version of Proposition 3.3.2 and of the minimizer of a weaker version of (3.11) discussed in Section 3.3.1.

The other track is on the task of robustly generating realistic images, where it is further applied to data augmentation in medical imaging or other fields [130, 131, 132, 133]. In the practical scenario, the training dataset for the generation tasks may be corrupted in the two possible cases. The first type is that the images themselves may be contaminated by (unknown) noises, which is usually known for adversarial attacks. Second type is that some of the images in the dataset with different structures might be considered as outliers. That is, they may belong to a different class of images, but wrongly classified and included in the training set [134]. Generative networks robust to the first type of corruption were recently studied in [135, 136, 137, 138]. However, to our best of knowledge, there is no previous work addressing the second type of corruption or the mixture of both types of corruptions. We are designing an end-to-end method to robustly generate high-quality images when the training set is contaminated according to the two different types of corruptions given that we are lacking of prior information of the labels for the training dataset or the types of corruptions. We expect our idea is not only applicable for robust images generation but also extended to robust audio generation.

In Chapter 4, we explain how symmetries in the forward processes can lead to difficulty—approximating highly oscillatory functions—in solving the resulting inverse problems by an end-to-end deep learning approach. Using the real and complex Gaussian PR problems as examples, we show how effective symmetry breaking can be performed to remove the above difficulties in learning, and we also verify the effectiveness of our scheme using extensive numerical experiments. We further extend our strategy to the nontrivial Fourier phase retrieval problem. In particular, we show through experiments that without carefully dealing with the symmetries, learning can be highly inefficient and the performance can be inferior to simple baseline methods. We also identify a basic principle for breaking symmetry and phrase the task as finding connected representative set for equivalence classes.

The task seems highly generic and only pertains to the certain topological and geometrical structure of the data space. It indicates our strategy is probably universal and we will investigate its extension to other nonlinear inverse problems. More precisely, we expect to develop a general strategy to identify the representative set of a system of inverse problems. In this case, the symmetries may not only come from each inverse problem but may also from the entangled system. Moreover, we expect to study the robustness theory when the observation is contaminated by (unknown) noises. We also remark that in some practical scenarios that we may not have prior information about symmetries and it may be challenging to identify all of them. A current work [139, 140] attempts to learn the equivalence classes of data points resulted from hidden symmetries. We expect to incorporate the idea and extend it to phase retrieval problems.

References

[1] Chieh-Hsin Lai, Dongmian Zou, and Gilad Lerman. Robust subspace recovery layer for unsupervised anomaly detection. In *International Conference on Learning Representations*, 2020.

[2] Chieh-Hsin Lai, Dongmian Zou, and Gilad Lerman. Novelty detection via robust variational autoencoding. *arXiv preprint arXiv:2006.05534*, 2020.

[3] Kshitij Tayal, Chieh-Hsin Lai, Raunak Manekar, Zhong Zhuang, Vipin Kumar, and Ju Sun. Unlocking inverse problems using deep learning: Breaking symmetries in phase retrieval. In *NeurIPS 2020 Workshop on Deep Learning and Inverse Problems*, 2020.

[4] Kshitij Tayal, Chieh-Hsin Lai, Vipin Kumar, and Ju Sun. Inverse problems, deep learning, and symmetry breaking. *arXiv preprint arXiv:2003.09077*, 2020.

[5] Markus Goldstein and Seiichi Uchida. A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data. *PloS one*, 11(4):e0152173, 2016.

[6] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):1–58, 2009.

[7] Yves Grandvalet, Johnny Mariéthoz, and Samy Bengio. A probabilistic interpretation of svms with an application to unbalanced classification. Technical report, IDIAP, 2005.

[8] Yanmin Sun, Andrew KC Wong, and Mohamed S Kamel. Classification of imbalanced data: A review. *International journal of pattern recognition and artificial intelligence*, 23(04):687–719, 2009.

[9] Seyda Ertekin, Jian Huang, Leon Bottou, and Lee Giles. Learning on the border: active learning in imbalanced data classification. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 127–136, 2007.

[10] Yuji Roh, Geon Heo, and Steven Euijong Whang. A survey on data collection for machine learning: a big data-ai integration perspective. *IEEE Transactions on Knowledge and Data Engineering*, 2019.

[11] Yan Xia, Xudong Cao, Fang Wen, Gang Hua, and Jian Sun. Learning discriminative reconstructions for unsupervised outlier removal. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1511–1519, 2015.

[12] Ramakrishnan Kannan, Hyenkyun Woo, Charu C. Aggarwal, and Haesun Park. Outlier detection for text data. In *Proceedings of the 2017 SIAM International Conference on Data Mining*, pages 489–497, 2017.

[13] Marco AF Pimentel, David A Clifton, Lei Clifton, and Lionel Tarassenko. A review of novelty detection. *Signal Processing*, 99:215–249, 2014.

[14] Raghavendra Chalapathy and Sanjay Chawla. Deep learning for anomaly detection: A survey. *arXiv preprint arXiv:1901.03407*, 2019.

[15] Pramuditha Perera, Poojan Oza, and Vishal M Patel. One-class classification: A survey. *arXiv preprint arXiv:2101.03064*, 2021.

[16] Xun Zhou, Sicong Cheng, Meng Zhu, Chengkun Guo, Sida Zhou, Peng Xu, Zhenghua Xue, and Weishi Zhang. A state of the art survey of data mining-based fraud detection and credit scoring. In *MATEC Web of Conferences*, volume 189, page 03002. EDP Sciences, 2018.

[17] Hugo Vieira Neto and Ulrich Nehmzow. Real-time automated visual inspection using mobile robots. *Journal of Intelligent and Robotic Systems*, 49(3):293–307, 2007.

[18] Qi Wei, Yinhao Ren, Rui Hou, Bibo Shi, Joseph Y Lo, and Lawrence Carin. Anomaly detection for medical images based on a one-class classification. In *Medical Imaging 2018: Computer-Aided Diagnosis*, volume 10575, page 105751M. International Society for Optics and Photonics, 2018.

[19] Mary M Moya and Don R Hush. Network constraints and multi-objective optimization for one-class classification. *Neural Networks*, 9(3):463–474, 1996.

[20] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[21] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing (4th Edition)*. Pearson, 2017.

[22] Pierre Comon. *Handbook of Blind Source Separation: Independent Component Analysis and Applications*. ACADEMIC PR INC, 2010.

[23] David Colton and Rainer Kress. *Inverse Acoustic and Electromagnetic Scattering Theory*. Springer New York, 2013.

[24] Gabor T. Herman. *Fundamentals of Computerized Tomography*. Springer London, 2009.

[25] Dara Entekhabi, Hajime Nakamura, and Eni G Njoku. Solving the inverse problem for soil moisture and temperature profiles by sequential assimilation of multifrequency remotely sensed observations. *IEEE Transactions on Geoscience and Remote Sensing*, 32(2):438–448, 1994.

[26] Rong Ge. *Provable Algorithms for Machine Learning Problems*. PhD thesis, Princeton University, 2013.

[27] Tomaso Poggio, Hrushikesh Mhaskar, Lorenzo Rosasco, Brando Miranda, and Qianli Liao. Why and when can deep-but not shallow-networks avoid the curse of

dimensionality: A review. *International Journal of Automation and Computing*, 14(5):503–519, mar 2017.

[28] Edmund Y. Lam and Joseph W. Goodman. Iterative statistical approach to blind image deconvolution. *Journal of the Optical Society of America A*, 17(7):1177, jul 2000.

[29] T. L. Tonellot and M. K. Broadhead. Sparse seismic deconvolution by method of orthogonal matching pursuit. In *72nd EAGE Conference and Exhibition incorporating SPE EUROPEC 2010*. EAGE Publications BV, jun 2010.

[30] Tamir Bendory, Robert Beinert, and Yonina C. Eldar. Fourier phase retrieval: Uniqueness and algorithms. In *Compressed Sensing and its Applications*, pages 55–91. Springer International Publishing, 2017.

[31] Mei-Ling Shyu, Shu-Ching Chen, Kanoksri Sarinnapakorn, and LiWu Chang. A novel anomaly detection scheme based on principal component classifier. In *Proc. ICDM Foundation and New Direction of Data Mining workshop, 2003*, pages 172–179, 2003.

[32] Gilad Lerman and Tyler Maunu. An overview of robust subspace recovery. *Proceedings of the IEEE*, 106(8):1380–1410, 2018.

[33] Namrata Vaswani and Praneeth Narayanamurthy. Static and dynamic robust PCA and matrix completion: A review. *Proceedings of the IEEE*, 106(8):1359–1379, 2018.

[34] John Wright, Arvind Ganesh, Shankar Rao, Yigang Peng, and Yi Ma. Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization. In *Advances in neural information processing systems*, pages 2080–2088, 2009.

[35] Chong Zhou and Randy C Paffenroth. Anomaly detection with robust deep autoencoders. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 665–674, 2017.

[36] Randy Paffenroth, Kathleen Kay, and Les Servi. Robust PCA for anomaly detection in cyber networks. *arXiv preprint arXiv:1801.01571*, 2018.

[37] Fernando De La Torre and Michael J Black. A framework for robust subspace learning. *International Journal of Computer Vision*, 54(1-3):117–142, 2003.

[38] Caglar Aytekin, Xingyang Ni, Francesco Cricri, and Emre Aksu. Clustering and unsupervised anomaly detection with l 2 normalized deep auto-encoder representations. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–6. IEEE, 2018.

[39] Bo Zong, Qi Song, Martin Renqiang Min, Wei Cheng, Cristian Lumezanu, Daeki Cho, and Haifeng Chen. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In *International Conference on Learning Representations*, 2018.

[40] Shuangfei Zhai, Yu Cheng, Weining Lu, and Zhongfei Zhang. Deep structured energy based models for anomaly detection. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48, pages 1100–1109. PMLR, 2016.

[41] Raghavendra Chalapathy, Aditya Krishna Menon, and Sanjay Chawla. Robust, deep and inductive anomaly detection. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 36–51. Springer, 2017.

[42] G. Alistair Watson. *Some Problems in Orthogonal Distance and Non-Orthogonal Distance Regression*. Defense Technical Information Center, 2001.

[43] Chris Ding, Ding Zhou, Xiaofeng He, and Hongyuan Zha. R1-PCA: rotational invariant l_1-norm principal component analysis for robust subspace factorization. In *Proceedings of the 23rd international conference on Machine learning*, pages 281–288. ACM, 2006.

[44] Teng Zhang, Arthur Szlam, and Gilad Lerman. Median K-flats for hybrid linear modeling with many outliers. In *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pages 234–241. IEEE, 2009.

[45] Michael McCoy and Joel A Tropp. Two proposals for robust PCA using semidefinite programming. *Electronic Journal of Statistics*, 5:1123–1160, 2011.

[46] Huan Xu, Constantine Caramanis, and Sujay Sanghavi. Robust PCA via outlier pursuit. *IEEE Trans. Information Theory*, 58(5):3047–3064, 2012.

[47] Gilad Lerman and Teng Zhang. l_p-recovery of the most significant subspace among multiple subspaces with outliers. *Constructive Approximation*, 40(3):329–385, 2014.

[48] Teng Zhang and Gilad Lerman. A novel M-estimator for robust PCA. *Journal of Machine Learning Research*, 15(1):749–808, 2014.

[49] Gilad Lerman, Michael B McCoy, Joel A Tropp, and Teng Zhang. Robust computation of linear models by convex relaxation. *Foundations of Computational Mathematics*, 15(2):363–410, 2015.

[50] Gilad Lerman and Tyler Maunu. Fast, robust and non-convex subspace recovery. *Information and Inference: A Journal of the IMA*, 7(2):277–336, 2017.

[51] Tyler Maunu, Teng Zhang, and Gilad Lerman. A well-tempered landscape for non-convex robust subspace recovery. *arXiv preprint arXiv:1706.03896*, 2017.

[52] Pan Ji, Tong Zhang, Hongdong Li, Mathieu Salzmann, and Ian Reid. Deep subspace clustering networks. In *Advances in Neural Information Processing Systems*, pages 24–33, 2017.

[53] José Lezama, Qiang Qiu, Pablo Musé, and Guillermo Sapiro. OLÉ: Orthogonal low-rank embedding-a plug and play geometric loss for deep learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8109–8118, 2018.

[54] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[55] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *Proceedings of the 34th International Conference on Machine Learning*, pages 214–223. PMLR, 2017.

[56] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference for Learning Representations*, 2015.

[57] Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. *Computer vision and Image understanding*, 106(1):59–70, 2007.

[58] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[59] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

[60] David Lewis. Reuters-21578 text categorization test collection. *Distribution 1.0, AT&T Labs-Research*, 1997.

[61] Ken Lang. Newsweeder: Learning to filter netnews. In *Proceedings of the Twelfth International Conference on Machine Learning*, pages 331–339, 1995.

[62] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pages 630–645. Springer, 2016.

[63] Anand Rajaraman and Jeffrey David Ullman. *Mining of massive datasets*. Cambridge University Press, 2011.

[64] Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. LOF: identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pages 93–104, 2000.

[65] Bernhard Schölkopf, Robert C Williamson, Alex J Smola, John Shawe-Taylor, and John C Platt. Support vector method for novelty detection. In *Advances in neural information processing systems*, pages 582–588, 2000.

[66] Mennatallah Amer, Markus Goldstein, and Slim Abdennadher. Enhancing one-class support vector machines for unsupervised anomaly detection. In *Proceedings of the ACM SIGKDD Workshop on Outlier Detection and Description*, pages 8–15. ACM, 2013.

[67] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation-based anomaly detection. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 6(1):3, 2012.

[68] Izhak Golan and Ran El-Yaniv. Deep anomaly detection using geometric transformations. In *Advances in Neural Information Processing Systems*, pages 9758–9769, 2018.

[69] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.

[70] Tyler Maunu and Gilad Lerman. Robust subspace recovery with adversarial outliers. *arXiv preprint arXiv:1904.03275*, 2019.

[71] Diederik P Kingma and Max Welling. Auto-encoding variational Bayes. In *2nd International Conference for Learning Representations. arXiv:1312.6114*, 2013.

[72] Jinwon An and Sungzoon Cho. Variational autoencoder based anomaly detection using reconstruction probability. *Special Lecture on IE*, 2(1), 2015.

[73] Aleksei Vasilev, Vladimir Golkov, Ilona Lipp, Eleonora Sgarlata, Valentina Tomassini, Derek K Jones, and Daniel Cremers. q-space novelty detection with variational autoencoders. *arXiv preprint arXiv:1806.02997*, 2018.

[74] Houssam Zenati, Chuan Sheng Foo, Bruno Lecouat, Gaurav Manek, and Vijay Ramaseshan Chandrasekhar. Efficient GAN-based anomaly detection, 2018.

[75] Mark Kliger and Shachar Fleishman. Novelty detection with GAN, 2018.

[76] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of Wasserstein GANs. In *Advances in Neural Information Processing Systems*, pages 5767–5777, 2017.

[77] Peter W Jones. Rectifiable sets and the traveling salesman problem. *Invent Math*, 102(1):1–15, 1990.

[78] Guy David, Stephen Semmes, G, and S Semmes David. *Analysis of and on uniformly rectifiable sets*, volume 38 of *Mathematical surveys and monographs*. American Mathematical Society, Providence, RI, 1993.

[79] Gilad Lerman. Quantifying curvelike structures of measures by using L_2 Jones quantities. *Comm. Pure Appl. Math.*, 56(9):1294–1365, 2003.

[80] Diederik P. Kingma and Max Welling. Auto-encoding variational Bayes. In *International Conference on Learning Representations*, 2014.

[81] Yoshua Bengio and Martin Monperrus. Non-local manifold tangent learning. In *Advances in Neural Information Processing Systems*, pages 129–136, 2005.

[82] Jarmo Ilonen, Pekka Paalanen, J-K Kamarainen, and H Kalviainen. Gaussian mixture pdf in one-class classification: computing and utilizing confidence values. In *18th International Conference on Pattern Recognition (ICPR'06)*, volume 2, pages 577–580. IEEE, 2006.

[83] Yingchao Xiao, Huangang Wang, Wenli Xu, and Junwu Zhou. Robust one-class SVM for fault detection. *Chemometrics and Intelligent Laboratory Systems*, 151:15 – 25, 2016.

[84] Kunzhe Wang and Haibin Lan. Robust support vector data description for novelty detection with contaminated data. *Engineering Applications of Artificial Intelligence*, 91:103554, 2020.

[85] Hansi Jiang, Haoyu Wang, Wenhao Hu, Deovrat Kakde, and Arin Chaudhuri. Fast incremental SVDD learning algorithm with the Gaussian kernel. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):3991–3998, Jul. 2019.

[86] Mei-Ling Shyu, Shu-Ching Chen, Kanoksri Sarinnapakorn, and LiWu Chang. A novel anomaly detection scheme based on principal component classifier. In *ICDM Foundation and New Direction of Data Mining workshop*, 2003.

[87] Mohammad Sabokrou, Mohammad Khalooei, Mahmood Fathy, and Ehsan Adeli. Adversarially learned one-class classifier for novelty detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3379–3388, 2018.

[88] Pramuditha Perera, Ramesh Nallapati, and Bing Xiang. OCGAN: One-class novelty detection using gans with constrained latent representations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2898–2906, 2019.

[89] Stanislav Pidhorskyi, Ranya Almohsen, and Gianfranco Doretto. Generative probabilistic novelty detection with adversarial autoencoders. In *Advances in neural information processing systems*, pages 6822–6833, 2018.

[90] Tangqing Li, Zheng Wang, Siying Liu, and Wen-Yan Lin. Deep unsupervised anomaly detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3636–3645, 2021.

[91] Michael I Jordan, Zoubin Ghahramani, Tommi S Jaakkola, and Lawrence K Saul. An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233, 1999.

[92] David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *Journal of the American statistical Association*, 112(518):859–877, 2017.

[93] Tal Daniel, Thanard Kurutach, and Aviv Tamar. Deep variational semi-supervised novelty detection. *arXiv preprint arXiv:1911.04971*, 2019.

[94] Haowen Xu, Wenxiao Chen, Nengwen Zhao, Zeyan Li, Jiahao Bu, Zhihan Li, Ying Liu, Youjian Zhao, Dan Pei, Yang Feng, et al. Unsupervised anomaly detection via variational auto-encoder for seasonal KPIs in web applications. In *Proceedings of the 2018 World Wide Web Conference*, pages 187–196, 2018.

[95] Chunkai Zhang, Shaocong Li, Hongye Zhang, and Yingyang Chen. VELC: A new variational autoencoder based model for time series anomaly detection. *arXiv preprint arXiv:1907.01702*, 2019.

[96] Adrian Pol, Victor Berger, Gianluca Cerminara, Cecile Germain, and Maurizio Pierini. Anomaly detection with conditional variational autoencoders. *HAL archive preprint hal-02396279*, 2019.

[97] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, and Ian Goodfellow. Adversarial autoencoders. In *International Conference on Learning Representations Workshop*, 2016.

[98] Ilya Tolstikhin, Olivier Bousquet, Sylvain Gelly, and Bernhard Schoelkopf. Wasserstein auto-encoders. In *International Conference on Learning Representations*, 2018.

[99] Emmanuel J. Candès, Xiaodong Li, Yi Ma, and John Wright. Robust principal component analysis? *J. ACM*, 58(3), June 2011.

[100] Tyler Maunu, Teng Zhang, and Gilad Lerman. A well-tempered landscape for non-convex robust subspace recovery. *Journal of Machine Learning Research*, 20(37):1–59, 2019.

[101] Bin Dai, Yu Wang, John Aston, Gang Hua, and David Wipf. Connections with robust pca and the role of emergent sparsity in variational autoencoder models. *The Journal of Machine Learning Research*, 19(1):1573–1614, 2018.

[102] Haleh Akrami, Anand A Joshi, Jian Li, and Richard M Leahy. Robust variational autoencoder. *arXiv preprint arXiv:1905.09961*, 2019.

[103] Simão Eduardo, Alfredo Nazábal, Christopher K. I. Williams, and Charles Sutton. Robust variational autoencoders for outlier detection and repair of mixed-type data. In *AISTATS*, 2020.

[104] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *Advances in Neural Information Processing Systems*, pages 7167–7177, 2018.

[105] Hendrik P. Lopuhaa and Peter J. Rousseeuw. Breakdown points of affine equivariant estimators of multivariate location and covariance matrices. *Ann. Statist.*, 19(1):229–248, 03 1991.

[106] Aarthi Reddy, Meredith Ordway-West, Melissa Lee, Matt Dugan, Joshua Whitney, Ronen Kahana, Brad Ford, Johan Muedsam, Austin Henslee, and Max Rao. Using gaussian mixture models to detect outliers in seasonal univariate network traffic. In *2017 IEEE Security and Privacy Workshops (SPW)*, pages 229–234. IEEE, 2017.

[107] Martial Agueh and Guillaume Carlier. Barycenters in the Wasserstein space. *SIAM Journal on Mathematical Analysis*, 43(2):904–924, 2011.

[108] Gabriel Peyré, Marco Cuturi, et al. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019.

[109] Yongxin Chen, Tryphon T Georgiou, and Allen Tannenbaum. Optimal transport for gaussian mixture models. *IEEE Access*, 7:6269–6278, 2018.

[110] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *2008 Eighth IEEE International Conference on Data Mining*, pages 413–422. IEEE, 2008.

[111] Katherine Heller, Krysta Svore, Angelos D Keromytis, and Salvatore Stolfo. One class support vector machines for detecting anomalous windows registry accesses. 2003.

[112] Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake VanderPlas, Arnaud Joly, Brian Holt, and Gaël Varoquaux. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pages 108–122, 2013.

[113] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256, 2010.

[114] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[115] Muhammad EH Chowdhury, Tawsifur Rahman, Amith Khandakar, Rashid Mazhar, Muhammad Abdul Kadir, Zaid Bin Mahbub, Khandakar R Islam, Muhammad Salman Khan, Atif Iqbal, Nasser Al-Emadi, et al. Can AI help in screening viral and COVID-19 pneumonia? *arXiv preprint arXiv:2003.13145*, 2020.

[116] Alex Krizhevsky et al. Learning multiple layers of features from tiny images. 2009.

[117] Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories. In *2004 conference on computer vision and pattern recognition workshop*, pages 178–178. IEEE, 2004.

[118] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.

[119] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. Mvtec ad–a comprehensive real-world dataset for unsupervised anomaly detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9592–9600, 2019.

[120] Michael T. McCann, Kyong Hwan Jin, and Michael Unser. Convolutional neural networks for inverse problems in imaging: A review. *IEEE Signal Processing Magazine*, 34(6):85–95, nov 2017.

[121] Alice Lucas, Michael Iliadis, Rafael Molina, and Aggelos K. Katsaggelos. Using deep neural networks for inverse problems in imaging: Beyond analytical methods. *IEEE Signal Processing Magazine*, 35(1):20–36, jan 2018.

[122] Simon Arridge, Peter Maass, Ozan Öktem, and Carola-Bibiane Schönlieb. Solving inverse problems using data-driven models. *Acta Numerica*, 28:1–174, may 2019.

[123] Yoav Shechtman, Yonina C Eldar, Oren Cohen, Henry Nicholas Chapman, Jianwei Miao, and Mordechai Segev. Phase retrieval with application to optical imaging: a contemporary overview. *IEEE signal processing magazine*, 32(3):87–109, 2015.

[124] Emmanuel J. Candès, Thomas Strohmer, and Vladislav Voroninski. PhaseLift: Exact and stable signal recovery from magnitude measurements via convex programming. *Communications on Pure and Applied Mathematics*, 66(8):1241–1274, nov 2012.

[125] Emmanuel J. Candes, Xiaodong Li, and Mahdi Soltanolkotabi. Phase retrieval via wirtinger flow: Theory and algorithms. *IEEE Transactions on Information Theory*, 61(4):1985–2007, apr 2015.

[126] Ju Sun, Qing Qu, and John Wright. A geometric analysis of phase retrieval. *Foundations of Computational Mathematics*, 18(5):1131–1198, aug 2017.

[127] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[128] Ayan Sinha, Justin Lee, Shuai Li, and George Barbastathis. Lensless computational imaging through deep learning. *Optica*, 4(9):1117, sep 2017.

[129] Z Zhuang, G Wang, Y Travadi, and J Sun. Phase retrieval via second-order nonsmooth optimization. *ICML workshop on Beyond First Order Methods in Machine Learning*, 2020.

[130] Ngoc-Trung Tran, Viet-Hung Tran, Ngoc-Bao Nguyen, Trung-Kien Nguyen, and Ngai-Man Cheung. On data augmentation for gan training. *IEEE Transactions on Image Processing*, 30:1882–1897, 2021.

[131] Antreas Antoniou, Amos Storkey, and Harrison Edwards. Data augmentation generative adversarial networks. *arXiv preprint arXiv:1711.04340*, 2017.

[132] Yu-Xiong Wang, Ross Girshick, Martial Hebert, and Bharath Hariharan. Low-shot learning from imaginary data. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7278–7286, 2018.

[133] Luis Perez and Jason Wang. The effectiveness of data augmentation in image classification using deep learning. *arXiv preprint arXiv:1712.04621*, 2017.

[134] Curtis G Northcutt, Anish Athalye, and Jonas Mueller. Pervasive label errors in test sets destabilize machine learning benchmarks. *arXiv preprint arXiv:2103.14749*, 2021.

[135] Stephan Zheng, Yang Song, Thomas Leung, and Ian Goodfellow. Improving the robustness of deep neural networks via stability training. In *Proceedings of the ieee conference on computer vision and pattern recognition*, pages 4480–4488, 2016.

[136] Steven Diamond, Vincent Sitzmann, Stephen Boyd, Gordon Wetzstein, and Felix Heide. Dirty pixels: Optimizing image classification architectures for raw sensor data. *arXiv preprint arXiv:1701.06487*, 2017.

[137] Ashish Bora, Eric Price, and Alexandros G Dimakis. Ambientgan: Generative models from lossy measurements. *ICLR*, 2(5):3, 2018.

[138] Takuhiro Kaneko and Tatsuya Harada. Noise robust generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8404–8414, 2020.

[139] Allan Zhou, Tom Knowles, and Chelsea Finn. Meta-learning symmetries by reparameterization. *arXiv preprint arXiv:2007.02933*, 2020.

[140] Mario Wieser, Sonali Parbhoo, Aleksander Wieczorek, and Volker Roth. Inverse learning of symmetries. *arXiv preprint arXiv:2002.02782*, 2020.

[141] John R Hershey and Peder A Olsen. Approximating the Kullback Leibler divergence between gaussian mixture models. In *2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07*, volume 4, pages IV–317. IEEE, 2007.

[142] Victor M Panaretos and Yoav Zemel. Statistical aspects of wasserstein distances. *Annual review of statistics and its application*, 6:405–431, 2019.

[143] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake VanderPlas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Edouard Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Appendix A

Supplementary proofs for Chapter 2

A.1 Proof of Proposition 2.8.1

Let $\boldsymbol{P}^\diamond$ be a minimizer of (2.10) and $(\boldsymbol{D}^\star, \boldsymbol{E}^\star)$ be a minimizer of (2.8). Since $\boldsymbol{P}^\diamond$ is an orthoprojector of rank d it can be written as $\boldsymbol{P}^\diamond = \boldsymbol{U}^\diamond \boldsymbol{U}^{\diamond\mathrm{T}}$, where $\boldsymbol{U}^\diamond \in \mathbb{R}^{D\times d}$, and thus

$$\sum_{t=1}^{N} \left\| \boldsymbol{x}^{(t)} - \boldsymbol{D}^\star \boldsymbol{E}^\star \boldsymbol{x}^{(t)} \right\|_2^p \leq \sum_{t=1}^{N} \left\| \boldsymbol{x}^{(t)} - \boldsymbol{U}^\diamond \boldsymbol{U}^{\diamond\mathrm{T}} \boldsymbol{x}^{(t)} \right\|_2^p = \sum_{t=1}^{N} \left\| \boldsymbol{x}^{(t)} - \boldsymbol{P}^\diamond \boldsymbol{x}^{(t)} \right\|_2^p . \quad \text{(A.1)}$$

Let $\mathscr{L}$ denote the column space of $\boldsymbol{D}^\star \boldsymbol{E}^\star$. Then by the property of orthoprojection

$$\left\| \boldsymbol{x}^{(t)} - \boldsymbol{D}^\star \boldsymbol{E}^\star \boldsymbol{x}^{(t)} \right\|_2 \geq \left\| \boldsymbol{x}^{(t)} - \boldsymbol{P}_{\mathscr{L}} \boldsymbol{x}^{(t)} \right\|_2 \text{ for } 1 \leq t \leq N \quad \text{(A.2)}$$

and consequently

$$\sum_{t=1}^{N} \left\| \boldsymbol{x}^{(t)} - \boldsymbol{D}^\star \boldsymbol{E}^\star \boldsymbol{x}^{(t)} \right\|_2^p \geq \sum_{t=1}^{N} \left\| \boldsymbol{x}^{(t)} - \boldsymbol{P}_{\mathscr{L}} \boldsymbol{x}^{(t)} \right\|_2^p \geq \sum_{t=1}^{N} \left\| \boldsymbol{x}^{(t)} - \boldsymbol{P}^\diamond \boldsymbol{x}^{(t)} \right\|_2^p . \quad \text{(A.3)}$$

The combination of (A.1) and (A.3) yields the following two equalities

$$\sum_{t=1}^{N} \left\| \boldsymbol{x}^{(t)} - \boldsymbol{P}_{\mathscr{L}} \boldsymbol{x}^{(t)} \right\|_2^p = \sum_{t=1}^{N} \left\| \boldsymbol{x}^{(t)} - \boldsymbol{P}^{\diamond} \boldsymbol{x}^{(t)} \right\|_2^p , \tag{A.4}$$

$$\sum_{t=1}^{N} \left\| \boldsymbol{x}^{(t)} - \boldsymbol{D}^{\star} \boldsymbol{E}^{\star} \boldsymbol{x}^{(t)} \right\|_2^p = \sum_{t=1}^{N} \left\| \boldsymbol{x}^{(t)} - \boldsymbol{P}_{\mathscr{L}} \boldsymbol{x}^{(t)} \right\|_2^p . \tag{A.5}$$

We note that (A.4) implies that $\boldsymbol{P}_{\mathscr{L}}$ is a minimizer of (2.10) (among all rank d orthoprojectors). We further note that (A.2) and (A.5) yield that for all $1 \leq t \leq N$

$$\left\| \boldsymbol{x}^{(t)} - \boldsymbol{D}^{\star} \boldsymbol{E}^{\star} \boldsymbol{x}^{(t)} \right\|_2 = \left\| \boldsymbol{x}^{(t)} - \boldsymbol{P}_{\mathscr{L}} \boldsymbol{x}^{(t)} \right\|_2 . \tag{A.6}$$

Since $\boldsymbol{D}^{\star} \boldsymbol{E}^{\star} \boldsymbol{x}^{(t)} \in \mathscr{L}$ and $\boldsymbol{P}_{\mathscr{L}}$ is an orthoprojector we conclude from (A.6) that

$$\boldsymbol{D}^{\star} \boldsymbol{E}^{\star} \boldsymbol{x}^{(t)} = \boldsymbol{P}_{\mathscr{L}} \boldsymbol{x}^{(t)} \quad \text{for } 1 \leq t \leq N. \tag{A.7}$$

We note that the definition of $(\boldsymbol{D}^{\star}, \boldsymbol{E}^{\star})$ implies that $\mathscr{L}$ (which is the column space of $\boldsymbol{D}^{\star} \boldsymbol{E}^{\star}$) is contained in the span of $\{\boldsymbol{x}^{(t)}\}_{t=1}^{N}$. We also recall that the dimension of the span of $\{\boldsymbol{x}^{(t)}\}_{t=1}^{N}$ is at least the dimension of $\mathscr{L}$, that is, d. Combining the latter facts with (A.7) we obtain that $\boldsymbol{D}^{\star} \boldsymbol{E}^{\star} = \boldsymbol{P}_{\mathscr{L}}$. This and the fact that $\boldsymbol{P}_{\mathscr{L}}$ is a minimizer of (2.10) (which was derived from (A.4)) concludes (2.9).

A.2 Proof of Proposition 2.8.2

We denote the subspace $\mathscr{L}$ in the left hand side of (2.13) by $\mathscr{L}^{\star}$ in order to distinguish it from the generic notation $\mathscr{L}$ for subspaces. Consider the random variable $X \sim \mu$,

Where μ is $\mathcal{N}(\boldsymbol{m}_X, \boldsymbol{\Sigma}_X)$. Fix $\pi \in \Pi(\boldsymbol{\mu}, \boldsymbol{\nu})$. We note that

$$\begin{aligned} & \mathbb{E}_{(X,Y)\sim\pi}\|X-Y\|_2^p \\ = & \int_{\mathbb{R}^D}\int_{\mathbb{R}^D} \|\boldsymbol{x}-\boldsymbol{y}\|_2^p \pi(\boldsymbol{x},\boldsymbol{y})\mathrm{dx}\ \mathrm{dy} \\ \geq & \min_{\dim\mathscr{L}=\mathrm{d}} \int_{\mathbb{R}^D} \mathrm{dist(x},\mathscr{L})^{\mathrm{p}} \int_{\mathbb{R}^{\mathrm{D}}} \pi(\mathrm{x},\mathrm{y})\mathrm{dy}\ \mathrm{dx} \\ = & \min_{\dim\mathscr{L}=\mathrm{d}} \int_{\mathbb{R}^D} \mathrm{dist(x},\mathscr{L})^{\mathrm{p}} \mu(\mathrm{x})\ \mathrm{dx} \\ = & \min_{\dim\mathscr{L}=\mathrm{d}} \mathbb{E}\|X-\boldsymbol{P}_{\mathscr{L}}X\|_2^p\ . \end{aligned} \tag{A.8}$$

The inequality in (A.8) holds since X is fixed and Y satisfies $(X,Y) \sim \pi$, so the distribution of Y is $\mathcal{N}(\boldsymbol{m}_Y, \boldsymbol{\Sigma}_Y)$. Therefore, almost surely, Y takes values in the d-dimensional affine subspace $\{\boldsymbol{y} \in \mathbb{R}^D : \boldsymbol{y} - \boldsymbol{m}_Y \in \mathrm{range}(\Sigma_\mathrm{Y})\}$. Furthermore, we note that equality in (A.8) is achieved when $Y = \boldsymbol{P}_{\mathscr{L}^\star}X$.

We conclude the proof by showing that

$$\boldsymbol{m}_X \in \mathscr{L}^\star. \tag{A.9}$$

Indeed, (A.9) implies that the orthogonal projection of $X \sim \mathcal{N}(\boldsymbol{m}_X, \boldsymbol{\Sigma}_X)$ onto $\mathscr{L}^\star$ results in a random variable with distribution $\boldsymbol{\nu}$ which is $\mathcal{N}(\boldsymbol{m}_X, \boldsymbol{P}_{\mathscr{L}^\star}\boldsymbol{\Sigma}_X\boldsymbol{P}_{\mathscr{L}^\star})$. By the above observation about the optimality of $Y = \boldsymbol{P}_{\mathscr{L}^\star}X$, the density of this distribution is the optimal solution of (2.12).

To prove (A.9), we assume without loss of generality that $\boldsymbol{m}_X = \boldsymbol{0}$. Denote the orthogonal projection of the origin onto the affine subspace $\mathscr{L}^\star$ by $\boldsymbol{m}_{\mathscr{L}^\star}$ and let $\mathscr{L}_0 = \mathscr{L}^\star - \boldsymbol{m}_{\mathscr{L}^\star}$. We need to show that $\mathscr{L}^\star = \mathscr{L}_0$, or equivalently, $\boldsymbol{m}_{\mathscr{L}^\star} = \boldsymbol{0}$. We note $\mathscr{L}_0$ is a linear subspace, $\boldsymbol{m}_{\mathscr{L}^\star}$ is orthogonal to $\mathscr{L}_0$ and thus there exists a rotation matrix $\boldsymbol{O}$ such that

$$\boldsymbol{O}\mathscr{L}_0 = \{(0,\cdots,0,z_{D-d+1},\cdots,z_D) :\ z_{D-d+1},\cdots z_D \in \mathbb{R}\}\ , \tag{A.10}$$

and

$$\boldsymbol{O}\boldsymbol{m}_{\mathscr{L}^\star} = (m_1,\cdots,m_{D-d},0,\cdots,0)\ . \tag{A.11}$$

For any $\boldsymbol{x} \in \mathbb{R}^D$ we note that $\mu(\boldsymbol{x}) = \mu(-\boldsymbol{x})$ since μ is Gaussian. Using this

observation, other basic observations and the notation $\boldsymbol{O}\boldsymbol{x} = (x'_1, \cdots, x'_D)$ we obtain that

$$
\begin{aligned}
& \operatorname{dist}(\mathrm{x}, \mathscr{L}^\star)^{\mathrm{p}} \mu(\mathrm{x}) + \operatorname{dist}(-\mathrm{x}, \mathscr{L}^\star)^{\mathrm{p}} \mu(-\mathrm{x}) \\
= \; & \left(\operatorname{dist}(\mathrm{x}, \mathscr{L}^\star)^{\mathrm{p}} + \operatorname{dist}(-\mathrm{x}, \mathscr{L}^\star)^{\mathrm{p}}\right) \mu(\boldsymbol{x}) \\
= \; & \left(\operatorname{dist}(\mathrm{O}\mathrm{x}, \mathrm{O}\mathscr{L}^\star)^{\mathrm{p}} + \operatorname{dist}(-\mathrm{O}\mathrm{x}, \mathrm{O}\mathscr{L}^\star)^{\mathrm{p}}\right) \mu(\boldsymbol{x}) \\
= \; & \left(\left(\sum_{i=1}^{D-d} (x'_i - m_i)^2\right)^{p/2} + \left(\sum_{i=1}^{D-d} (-x'_i - m_i)^2\right)^{p/2}\right) \mu(\boldsymbol{x}) \\
= \; & \left(\left(\sum_{i=1}^{D-d} (x'_i - m_i)^2\right)^{p/2} + \left(\sum_{i=1}^{D-d} (x'_i + m_i)^2\right)^{p/2}\right) \mu(\boldsymbol{x}) \\
\geq \; & 2 \left(\sum_{i=1}^{D-d} {x'_i}^2\right)^{p/2} \mu(\boldsymbol{x}) \qquad \text{(A.12)} \\
= \; & 2 \operatorname{dist}(\mathrm{O}\mathrm{x}, \mathrm{O}\mathscr{L}_0)^{\mathrm{p}} \mu(\mathrm{x}) \\
= \; & 2 \operatorname{dist}(\mathrm{x}, \mathscr{L}_0)^{\mathrm{p}} \mu(\mathrm{x}) \\
= \; & \left(\operatorname{dist}(\mathrm{x}, \mathscr{L}_0)^{\mathrm{p}} + \operatorname{dist}(-\mathrm{x}, \mathscr{L}_0)^{\mathrm{p}}\right) \mu(\boldsymbol{x}) \\
= \; & \operatorname{dist}(\mathrm{x}, \mathscr{L}_0)^{\mathrm{p}} \mu(\mathrm{x}) + \operatorname{dist}(-\mathrm{x}, \mathscr{L}_0)^{\mathrm{p}} \mu(-\mathrm{x}) \; .
\end{aligned}
$$

The inequality in (A.12) follows from the fact that for $p \geq 1$, the function $\|\cdot\|_2^p$ is convex as it is a composition of the convex function $\|\cdot\|_2 : \mathbb{R}^d \to \mathbb{R}_+$ and the increasing convex function $(\cdot)^p : \mathbb{R}_+ \to \mathbb{R}_+$. Equality is achieved in (A.12) if $m_i = 0$ for $i = 1, \cdots, D-d$, that is, $\mathscr{L}^\star = \mathscr{L}_0$.

Integrating the left and right hand sides of (A.12) over $\mathbb{R}^D$ results in

$$
\int_{\mathbb{R}^D} \operatorname{dist}(\mathrm{x}, \mathscr{L}^\star)^{\mathrm{p}} \mu(\mathrm{x}) \mathrm{dx} \geq \int_{\mathbb{R}^{\mathrm{D}}} \operatorname{dist}(\mathrm{x}, \mathscr{L}_0)^{\mathrm{p}} \mu(\mathrm{x}) \mathrm{dx} \; . \qquad \text{(A.13)}
$$

Since $\mathscr{L}^\star$ is a minimizer among all affine subspaces of rank d of $\int_{\mathbb{R}^D} \operatorname{dist}(\mathrm{x}, \mathscr{L})^{\mathrm{p}} \mu(\mathrm{x}) \, \mathrm{dx} = \mathbb{E}\|\mathrm{X} - \mathrm{P}_{\mathscr{L}}\mathrm{X}\|_2^{\mathrm{p}}$, equality is obtained in (A.13). Consequently, equality is obtained, almost everywhere, in (A.12). Therefore, $\mathscr{L}^\star = \mathscr{L}_0$ and the claim is proved.

A.3 Proof of Proposition 2.8.3

Let $\boldsymbol{A}^\star$ be an optimizer of (2.14) and $\boldsymbol{P}^\star$ denote the orthogonal projection onto the range of $\boldsymbol{A}^{\star\mathrm{T}}\boldsymbol{A}^\star$. Note that $\boldsymbol{P}^\star$ can be written as $\tilde{\boldsymbol{A}}^{\mathrm{T}}\tilde{\boldsymbol{A}}$, where $\tilde{\boldsymbol{A}}$ is a $d \times D$ matrix composed of an orthonormal basis of the range of $\boldsymbol{P}^\star$. Therefore, being an optimum of (2.14), $\boldsymbol{A}^\star$ satisfies

$$\left\|\boldsymbol{z}^{(t)} - \boldsymbol{P}^\star\boldsymbol{z}^{(t)}\right\|_2 \geq \left\|\boldsymbol{z}^{(t)} - \boldsymbol{A}^{\star\mathrm{T}}\boldsymbol{A}^\star\boldsymbol{z}^{(t)}\right\|_2 , \quad t = 1, \cdots, N \ . \tag{A.14}$$

On the other hand, the definition of orthogonal projection implies that

$$\left\|\boldsymbol{z}^{(t)} - \boldsymbol{P}^\star\boldsymbol{z}^{(t)}\right\|_2 \leq \left\|\boldsymbol{z}^{(t)} - \boldsymbol{A}^{\star\mathrm{T}}\boldsymbol{A}^\star\boldsymbol{z}^{(t)}\right\|_2 , \quad t = 1, \cdots, N \ . \tag{A.15}$$

That is, equality is obtained in (A.14) and (A.15). This equality and the fact that $\boldsymbol{P}^\star$ is a projection on the range of $\boldsymbol{A}^{\star\mathrm{T}}\boldsymbol{A}^\star$ imply that

$$\boldsymbol{P}^\star\boldsymbol{z}^{(t)} = \boldsymbol{A}^{\star\mathrm{T}}\boldsymbol{A}^\star\boldsymbol{z}^{(t)} , \quad t = 1, \cdots, N \ . \tag{A.16}$$

Since $\{\boldsymbol{z}^{(t)}\}_{t=1}^N$ spans $\mathbb{R}^D$, (A.16) results in

$$\boldsymbol{P}^\star = \boldsymbol{A}^{\star\mathrm{T}}\boldsymbol{A}^\star \ , \tag{A.17}$$

which further implies that

$$\boldsymbol{A}^\star\boldsymbol{A}^{\star\mathrm{T}}\boldsymbol{A}^\star = \boldsymbol{A}^\star\boldsymbol{P}^\star = \boldsymbol{A}^\star \ . \tag{A.18}$$

Combining this observation ($\boldsymbol{A}^\star\boldsymbol{A}^{\star\mathrm{T}}\boldsymbol{A}^\star = \boldsymbol{A}^\star$) with the constraint that $\boldsymbol{A}^\star$ has a full rank, we conclude that $\boldsymbol{A}^\star\boldsymbol{A}^{\star\mathrm{T}} = \boldsymbol{I}_d$.

Appendix B

Supplementary proofs for Chapter 3

B.1 Proof of Proposition 3.3.1

Recall that $\boldsymbol{\mu}_0 \in \mathbb{R}^K$ is the mean of the prior Gaussian, $\epsilon > 0$ is the fixed separation parameter for the means of the two modes and $\eta > 1/2$ is the fixed mixture parameter. For $i = 0,\ 1,\ 2$, we denote the Gaussian probability distribution by $\mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$. Since in our setting $\boldsymbol{\Sigma}_0 = \boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2$, we denote the common covariance matrix in $\mathcal{S}^K_{++}$ by $\boldsymbol{\Sigma}$. That is, $\boldsymbol{\Sigma} = \boldsymbol{\Sigma}_i$ for $i = 0, 1, 2$.

We first analyze the solution of (3.11) with $\mathcal{R} = W_p$, where $p \geq 1$, and then analyze the solution of (3.11) with $\mathcal{R} = KL$.

The case $\boldsymbol{\mathcal{R} = W_p, p \geq 1}$: We follow the next three steps to prove that the minimizer of (3.11) satisfies $\boldsymbol{\mu}_1 = \boldsymbol{\mu}_0$.

Step I: We prove that

$$W_p(\nu_i, \nu_0) \equiv W_p(\mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}), \mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{\Sigma})) = \|\boldsymbol{\mu}_i - \boldsymbol{\mu}_0\|_2 \ \text{ for } p \geq 1 \text{ and } i = 1, 2\ . \quad \text{(B.1)}$$

First, we note that using the definition of $W_p, p \geq 1$ and the common notation

$\Pi(\nu_i, \nu_0)$ for the distribution on $\mathbb{R}^K \times \mathbb{R}^K$ with marginals ν_i and ν_0

$$\begin{aligned} W_p^p(\nu_i, \nu_0) &= \inf_{\pi \in \Pi(\nu_i, \nu_0)} \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim\pi} \|\boldsymbol{x} - \boldsymbol{y}\|_2^p \\ &\geq \inf_{\pi \in \Pi(\nu_i, \nu_0)} \left\| \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim\pi} \boldsymbol{x} - \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim\pi} \boldsymbol{y} \right\|_2^p \\ &= \|\boldsymbol{\mu}_i - \boldsymbol{\mu}_0\|_2^p \ , \end{aligned} \tag{B.2}$$

where the inequality follows the fact that $\|.\|_2^p$ is convex and from Jensen's inequality.

On the other hand, for $i = 1$ or $i = 2$, let $\boldsymbol{x}^*$ be an arbitrary random vector with distribution ν_i, and let $\boldsymbol{y}^* = \boldsymbol{x}^* - \boldsymbol{\mu}_i + \boldsymbol{\mu}_0$. The distribution of $\boldsymbol{y}^*$ is Gaussian with mean $\boldsymbol{\mu}_0$ and covariance $\boldsymbol{\Sigma}_i$, that is, this distribution is ν_0. Let π^* be the joint distribution of the random variables $\boldsymbol{x}^*$ and $\boldsymbol{y}^*$. We note that π^* is in $\Pi(\nu_i, \nu_0)$ and that

$$\mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim\pi^*} \|\boldsymbol{x} - \boldsymbol{y}\|_2^p = \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim\pi^*} \|\boldsymbol{\mu}_i - \boldsymbol{\mu}_0\|_2^p = \|\boldsymbol{\mu}_i - \boldsymbol{\mu}_0\|_2^p \ .$$

Therefore,

$$W_p^p(\nu_i, \nu_0) = \inf_{\pi \in \Pi(\nu_i, \nu_0)} \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim\pi} \|\boldsymbol{x} - \boldsymbol{y}\|_2^p \leq \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim\pi^*} \|\boldsymbol{x} - \boldsymbol{y}\|_2^p = \|\boldsymbol{\mu}_i - \boldsymbol{\mu}_0\|_2^p \ . \tag{B.3}$$

The combination of (B.2) and (B.3) immediately yields (B.1).

Step II: We prove that (3.11) with $\mathcal{R} = W_p$, $p \geq 1$, is equivalent to

$$\min_{\substack{\boldsymbol{\mu}_1, \boldsymbol{\mu}_2 \in \mathbb{R}^K; \\ \text{s.t. } \boldsymbol{\mu}_0, \boldsymbol{\mu}_1, \boldsymbol{\mu}_2 \text{:colinear} \\ \& \|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\|_2 \geq \epsilon}} \eta \|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0\|_2 + (1-\eta) \|\boldsymbol{\mu}_2 - \boldsymbol{\mu}_0\|_2. \tag{B.4}$$

We first note that (3.11) with $\mathcal{R} = W_p$, $p \geq 1$ is equivalent to

$$\min_{\substack{\boldsymbol{\mu}_1, \boldsymbol{\mu}_2 \in \mathbb{R}^K \\ \text{s.t. } \|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\|_2 \geq \epsilon}} \eta \|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0\|_2 + (1-\eta) \|\boldsymbol{\mu}_2 - \boldsymbol{\mu}_0\|_2. \tag{B.5}$$

Indeed, this is a direct consequence of the expression derived in step I for $\mathcal{R}$ in this case. It is thus left to show that if $\boldsymbol{\mu}_1'$, $\boldsymbol{\mu}_2' \in \mathbb{R}^K$ minimize (B.5), then we can construct $\widetilde{\boldsymbol{\mu}_1'}$, $\widetilde{\boldsymbol{\mu}_2'} \in \mathbb{R}^K$ that are colinear with $\boldsymbol{\mu}_0$ and also minimize (B.5).

For any $\boldsymbol{\mu_1}$ and $\boldsymbol{\mu_2}$ in $\mathbb{R}^K$ with $\|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\|_2 \geq \epsilon$ and for the given $\boldsymbol{\mu}_0 \in \mathbb{R}^K$, we

define $\tilde{\boldsymbol{\mu}}_0$, $\tilde{\boldsymbol{\mu}}_1$ and $\tilde{\boldsymbol{\mu}}_2 \in \mathbb{R}^K$ and demonstrate them in Figure B.1. The point $\tilde{\boldsymbol{\mu}}_0$ is the projection of $\boldsymbol{\mu}_0$ onto $\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2$ and $\tilde{\boldsymbol{\mu}}_i := \boldsymbol{\mu}_i + \boldsymbol{\mu}_0 - \tilde{\boldsymbol{\mu}}_0$ for $i = 1, 2$. We observe the following properties, which can be proved by direct calculation, though Figure B.1 also clarifies them:

$$\|\boldsymbol{\mu}_i - \boldsymbol{\mu}_0\|_2 \geq \|\tilde{\boldsymbol{\mu}}_i - \boldsymbol{\mu}_0\|_2 \text{ for } i = 1, 2,$$

and consequently,

$$\eta\|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0\|_2 + (1-\eta)\|\boldsymbol{\mu}_2 - \boldsymbol{\mu}_0\|_2 \geq \eta\|\tilde{\boldsymbol{\mu}}_1 - \boldsymbol{\mu}_0\|_2 + (1-\eta)\|\tilde{\boldsymbol{\mu}}_2 - \boldsymbol{\mu}_0\|_2; \tag{B.6}$$

$$\|\tilde{\boldsymbol{\mu}}_1 - \tilde{\boldsymbol{\mu}}_2\|_2 = \|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\|_2 \geq \epsilon; \tag{B.7}$$

and

$$\tilde{\boldsymbol{\mu}}_1,\ \tilde{\boldsymbol{\mu}}_2,\ \text{ and } \boldsymbol{\mu}_0 \text{ are colinear.} \tag{B.8}$$

Clearly, the combination of (B.6), (B.7) and (B.8) concludes the proof of step II. That is, it implies that if $\boldsymbol{\mu}_1'$, $\boldsymbol{\mu}_2' \in \mathbb{R}^K$ minimize (B.5), then $\widetilde{\boldsymbol{\mu}_1'}$ and $\widetilde{\boldsymbol{\mu}_2'}$ defined above are colinear with $\boldsymbol{\mu}_0$ and also minimize (B.5).

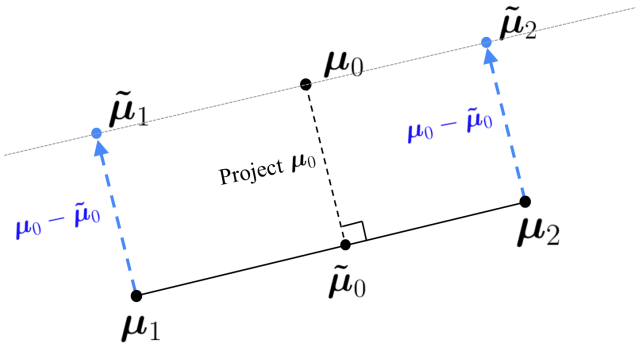


Figure B.1: Illustration of the points $\tilde{\boldsymbol{\mu}}_0$, $\tilde{\boldsymbol{\mu}}_1$ and $\tilde{\boldsymbol{\mu}}_2$ and their properties.

Step III: We directly solve (B.4) and consequently (3.11) with $\mathcal{R} = W_p$, $p \geq 1$. Due to the colinearity constraint in (3.11), we can write

$$\boldsymbol{\mu_0} = (1+t)\boldsymbol{\mu_1} - t\boldsymbol{\mu_2} \text{ for } t \in \mathbb{R}. \tag{B.9}$$

The objective function in (B.4) can then be written as

$$\|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\|_2 \left(\eta|t| + (1-\eta)|1+t|\right) \geq \epsilon \left(\eta|t| + (1-\eta)|1+t|\right),$$

where equality is achieved if and only if $\|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\|_2 = \epsilon$. We thus define $r(t) = \eta|t| + (1-\eta)|1+t|$ and note that

$$r(t) = \begin{cases} t + (1-\eta), & t \geq 0 \\ (1-2\eta)t + (1-\eta), & 0 \geq t \geq -1 \\ -t + (\eta - 1), & -1 \geq t \end{cases}$$

and its derivative is

$$r'(t) = \begin{cases} 1, & t > 0 \\ 1 - 2\eta, & 0 > t > -1 \\ -1, & -1 > t. \end{cases}$$

The above expressions for r and r' and the assumption that $\eta > 1/2$ imply that $r(t)$ is increasing when $t > 0$, decreasing when $t < 0$ and $r(0) = 1 - \eta < \eta = r(1)$. Thus r has a global minimum at $t = 0$. Hence, it follows from (B.9) that the minimizer of (3.11), and equivalently (3.11) with $\mathcal{R} = W_p$, $p \geq 1$ satisfies $\boldsymbol{\mu}_1 = \boldsymbol{\mu}_0$.

The case $\boldsymbol{\mathcal{R} = KL}$: We prove that the solution of (3.11) with $\mathcal{R} = KL$ satisfies $\boldsymbol{\mu}_0 = \eta\boldsymbol{\mu}_1 + (1-\eta)\boldsymbol{\mu}_2$. We practically follow similar steps as the proof above.

Step I: We derive an expression for $KL(\nu_i||\nu_0)$, where $i = 1, 2$. We use the following general formula, which holds for the case where $\boldsymbol{\Sigma}_0$, $\boldsymbol{\Sigma}_1$ and $\boldsymbol{\Sigma}_2$ are general covariance matrices in $\mathcal{S}_{++}^K$ (see e.g., (2) in [141]):

$$KL(\nu_i||\nu_0) = \frac{1}{2}\left(\log\frac{\det\boldsymbol{\Sigma}_0}{\det\boldsymbol{\Sigma}_i} - K + \operatorname{tr}(\boldsymbol{\Sigma}_0^{-1}\boldsymbol{\Sigma}_i) + (\boldsymbol{\mu}_i - \boldsymbol{\mu}_0)^{\mathrm{T}}\boldsymbol{\Sigma}_0^{-1}(\boldsymbol{\mu}_i - \boldsymbol{\mu}_0)\right). \quad \text{(B.10)}$$

Since in our setting $\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = \boldsymbol{\Sigma}$, this expression has the simpler form:

$$KL(\nu_i||\nu_0) = \frac{1}{2}(\boldsymbol{\mu}_i - \boldsymbol{\mu}_0)^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_i - \boldsymbol{\mu}_0).$$

Step II: We reformulate the optimization problem. The above step imples that (3.11) with $\mathcal{R} = KL$ can be written as

$$\min_{\|\boldsymbol{\mu}_1-\boldsymbol{\mu}_2\|_2\geq\epsilon} \quad \eta(\boldsymbol{\mu}_1-\boldsymbol{\mu}_0)^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1-\boldsymbol{\mu}_0)+(1-\eta)(\boldsymbol{\mu}_2-\boldsymbol{\mu}_0)^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_2-\boldsymbol{\mu}_0),$$

or equivalently,

$$\min_{\|\boldsymbol{\mu}_1-\boldsymbol{\mu}_2\|_2\geq\epsilon} \quad \eta\left\|\boldsymbol{\Sigma}^{-\frac{1}{2}}(\boldsymbol{\mu}_1-\boldsymbol{\mu}_0)\right\|_2^2+(1-\eta)\left\|\boldsymbol{\Sigma}^{-\frac{1}{2}}(\boldsymbol{\mu}_2-\boldsymbol{\mu}_0)\right\|_2^2. \tag{B.11}$$

We express the eigenvalue decomposition of $\boldsymbol{\Sigma}^{-1}$ as $\boldsymbol{\Sigma}^{-1}=\boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^{\mathrm{T}}$, where $\boldsymbol{\Lambda}\in\mathcal{S}_+^K$, and $\boldsymbol{U}$ is an orthogonal matrix. Applying the change of variables $\boldsymbol{\mu}_i^{'}=\boldsymbol{\Lambda}^{\frac{1}{2}}\boldsymbol{U}^{\mathrm{T}}\boldsymbol{\mu}_i$

for $i=0,\,1,\,2$, we rewrite (B.11) as

$$\min_{\left\|\boldsymbol{\mu}_1^{'}-\boldsymbol{\mu}_2^{'}\right\|_2\geq\epsilon} \quad \eta\left\|\boldsymbol{\mu}_1^{'}-\boldsymbol{\mu}_0^{'}\right\|_2^2+(1-\eta)\left\|\boldsymbol{\mu}_2^{'}-\boldsymbol{\mu}_0^{'}\right\|_2^2. \tag{B.12}$$

At last, applying the same colinearity argument as above (supported by Figure B.1)

we conclude the following equivalent formulation of (B.12):

$$\min_{\substack{\boldsymbol{\mu}_0^{'},\boldsymbol{\mu}_1^{'},\boldsymbol{\mu}_2^{'}\text{ are colinear}\\ \&\left\|\boldsymbol{\mu}_1^{'}-\boldsymbol{\mu}_2^{'}\right\|_2\geq\epsilon}} \quad \eta\left\|\boldsymbol{\mu}_1^{'}-\boldsymbol{\mu}_0^{'}\right\|_2^2+(1-\eta)\left\|\boldsymbol{\mu}_2^{'}-\boldsymbol{\mu}_0^{'}\right\|_2^2 \tag{B.13}$$

Step III: We directly solve (B.13). Due to the colinearity constraint, we can write

$$\boldsymbol{\mu_0^{'}}=(1+t)\boldsymbol{\mu_1^{'}}-t\boldsymbol{\mu_2^{'}} \;\text{ for } t\in\mathbb{R} \tag{B.14}$$

and express the objective function of (B.13) as

$$\left\|\boldsymbol{\mu}_1^{'}-\boldsymbol{\mu}_2^{'}\right\|_2^2\left(\eta t^2+(1-\eta)(1+t)^2\right)\geq\epsilon^2\left(\eta t^2+(1-\eta)(1+t)^2\right),$$

where equality is achieved if and only if $\|\boldsymbol{\mu}_1'-\boldsymbol{\mu}_2'\|_2=\epsilon$. We thus define $r(t)=\eta t^2+(1-\eta)(1+t)^2$ and note that $r'(t)=2(t+(1-\eta))$ and $r''(t)=2$, and thus conclude that $r(t)$ obtains its global minimum at $t=\eta-1$. This observation and (B.14) imply that the minimizers $\boldsymbol{\mu_1}$ and $\boldsymbol{\mu_2}$ of (3.11) with $\mathcal{R}=KL$ satisfy $\boldsymbol{\mu_0}=\eta\boldsymbol{\mu_1}+(1-\eta)\boldsymbol{\mu_2}$.

B.2 Proof of Proposition 3.3.2

We follow the same steps of the proof of Proposition 3.3.1 **Step I:** We immediately verify the formula

$$W_2(\mathcal{N}(\boldsymbol{\mu}_i,\boldsymbol{\Sigma}_i),\mathcal{N}(\mathbf{0},\boldsymbol{I}))=\sqrt{\|\boldsymbol{\mu}_i\|_2^2+\left\|\boldsymbol{\Sigma}_i^{\frac{1}{2}}-\boldsymbol{I}\right\|_F^2}\ \text{ for } i=1,2. \tag{B.15}$$

We use the following general formula, which holds for the case where $\boldsymbol{\Sigma}_0$, $\boldsymbol{\Sigma}_1$ and $\boldsymbol{\Sigma}_2$ are general covariance matrices in $\mathcal{S}_+^K$ (see e.g., (4) in [142]):

$$W_2^2(\mathcal{N}(\boldsymbol{\mu}_i,\boldsymbol{\Sigma}_i),\mathcal{N}(\boldsymbol{\mu}_0,\boldsymbol{\Sigma}_0))=\|\boldsymbol{\mu}_i-\boldsymbol{\mu}_0\|_2^2+\mathrm{tr}(\boldsymbol{\Sigma}_i+\boldsymbol{\Sigma}_0-2(\boldsymbol{\Sigma}_i^{\frac{1}{2}}\boldsymbol{\Sigma}_0\boldsymbol{\Sigma}_i^{\frac{1}{2}})^{\frac{1}{2}}),\ i=1,2\ . \tag{B.16}$$

Indeed, (B.15) is obtained as a direct consequence of (B.16) using the identity

$$\mathrm{tr}\left(\boldsymbol{\Sigma}_i+\boldsymbol{I}-2\boldsymbol{\Sigma}_i^{\frac{1}{2}}\right)=\mathrm{tr}\left(\left(\boldsymbol{\Sigma}_i^{\frac{1}{2}}-\boldsymbol{I}\right)^2\right)=\left\|\boldsymbol{\Sigma}_i^{\frac{1}{2}}-\boldsymbol{I}\right\|_F^2.$$

Step II: We reformulate the underlying minimization problem in two different stages. We first claim that the minimizer of (3.11) with $\mathcal{R}=W_2$ and the constraint that $\boldsymbol{\Sigma}_1$ is of rank κ and $\boldsymbol{\Sigma}_2$ is of rank K can be expressed as the minimizer of

$$\min_{\substack{\boldsymbol{\mu}_1,\boldsymbol{\mu}_2\in\mathbb{R}^K\text{s.t. }\|\boldsymbol{\mu}_1-\boldsymbol{\mu}_2\|_2=\epsilon,\\ \boldsymbol{\Sigma}_1,\boldsymbol{\Sigma}_2\ \text{diagonal in }\mathbb{R}^{K\times K}\\ \&\ \mathrm{rank}(\boldsymbol{\Sigma}_1)=\kappa,\ \mathrm{rank}(\boldsymbol{\Sigma}_2)=K}} \eta\sqrt{\|\boldsymbol{\mu}_1\|_2^2+\left\|\boldsymbol{\Sigma}_1^{\frac{1}{2}}-\boldsymbol{I}\right\|_F^2}+(1-\eta)\sqrt{\|\boldsymbol{\mu}_2\|_2^2+\left\|\boldsymbol{\Sigma}_2^{\frac{1}{2}}-\boldsymbol{I}\right\|_F^2}. \tag{B.17}$$

In view of (3.11) and (B.15) we only need to prove that the minimizer of (B.17) is the same if one removes the constraint that $\boldsymbol{\Sigma}_1$ and $\boldsymbol{\Sigma}_2$ are both diagonal matrices and require instead that they are in $\in\mathcal{S}_+^K$. This is easy to show. Indeed, if for $i=1$ or $i=2$, $\boldsymbol{\Sigma}_i\in\mathcal{S}_+^K$, then it can be diagonalized as follows: $\boldsymbol{\Sigma}_i=\boldsymbol{U}_i^{\mathrm{T}}\boldsymbol{\Lambda}_i\boldsymbol{U}_i$, where $\boldsymbol{\Lambda}_i\in\mathcal{S}_+^K$ is diagonal and $\boldsymbol{U}_i$ is orthogonal. Hence, $\boldsymbol{\Sigma}_i^{\frac{1}{2}}=\boldsymbol{U}_i^{\mathrm{T}}\boldsymbol{\Lambda}_i^{\frac{1}{2}}\boldsymbol{U}_i$ and $\left\|\boldsymbol{\Sigma}_i^{\frac{1}{2}}-\boldsymbol{I}\right\|_F^2=\left\|\boldsymbol{U}_i^{\mathrm{T}}\boldsymbol{\Lambda}_i^{\frac{1}{2}}\boldsymbol{U}_i-\boldsymbol{I}\right\|_F^2=\left\|\boldsymbol{U}_i^{\mathrm{T}}(\boldsymbol{\Lambda}_i^{\frac{1}{2}}-\boldsymbol{I})\boldsymbol{U}_i\right\|_F^2=\left\|\boldsymbol{\Lambda}_i^{\frac{1}{2}}-\boldsymbol{I}\right\|_F^2$. Consequently,

$$W_2(\mathcal{N}(\boldsymbol{\mu}_i,\boldsymbol{\Sigma}_i),\mathcal{N}(\mathbf{0},\boldsymbol{I}))=W_2(\mathcal{N}(\boldsymbol{\mu}_i,\boldsymbol{\Lambda}_i),\mathcal{N}(\mathbf{0},\boldsymbol{I}))\ \text{ for } i=1,2\ ,$$

and the above claim is concluded.

Next, we vectorize the minimization problem in (B.17) as follows. We denote by $\mathbb{R}_+$ the set of positive real numbers. Let $\boldsymbol{b}$ be a general vector in $\mathbb{R}_+^K$, $\boldsymbol{a'}$ be a general vector in $\mathbb{R}_+^\kappa$ and $\boldsymbol{a} := (\boldsymbol{a'}; \mathbf{0}_{K-\kappa}) \in \mathbb{R}^K$. Given, the constraints on $\boldsymbol{\Sigma}_1$ and $\boldsymbol{\Sigma}_2$, we can parametrize the diagonal elements of $\boldsymbol{\Sigma}_1^{\frac{1}{2}}$ and $\boldsymbol{\Sigma}_2^{\frac{1}{2}}$ by $\boldsymbol{a}$ and $\boldsymbol{b}$, that is, we set $\boldsymbol{\Sigma}_1^{\frac{1}{2}} = \mathrm{diag}(\boldsymbol{a})$ and $\boldsymbol{\Sigma}_2^{\frac{1}{2}} = \mathrm{diag}(\boldsymbol{b})$. The objective function of (B.17) can then be written as

$$\eta\sqrt{\|\boldsymbol{\mu_1}\|_2^2 + \|\boldsymbol{a} - \mathbf{1}_K\|_2^2} + (1-\eta)\sqrt{\|\boldsymbol{\mu_2}\|_2^2 + \|\boldsymbol{b} - \mathbf{1}_K\|_2^2}.$$

Combining this last expression and the same colinearity argument as in §B.1 (supported by Figure B.1), (B.17) is equivalent to

$$\min_{\substack{\boldsymbol{\mu}_1,\boldsymbol{\mu}_2\in\mathbb{R}^K,\ \boldsymbol{b}\in\mathbb{R}_+^K,\ \boldsymbol{a'}\in\mathbb{R}_+^\kappa,\ \boldsymbol{a}=(\boldsymbol{a'};\mathbf{0}_{K-\kappa}),\\ (\boldsymbol{\mu}_1;\boldsymbol{a}),(\boldsymbol{\mu}_2;\boldsymbol{b}),(\mathbf{0}_K;\mathbf{1}_K)\text{ are colinear}\\ \&\|\boldsymbol{\mu}_1-\boldsymbol{\mu}_2\|_2=\epsilon}} \eta\|(\boldsymbol{\mu}_1;\boldsymbol{a}) - (\mathbf{0}_K;\mathbf{1}_K)\|_2 + (1-\eta)\|(\boldsymbol{\mu}_2;\boldsymbol{b}) - (\mathbf{0}_K;\mathbf{1}_K)\|_2. \tag{B.18}$$

Step III: We solve (B.18). By the colinearity constraint, we can write $(\mathbf{0}_K;\mathbf{1}_K) = u(\boldsymbol{\mu}_2;\boldsymbol{b}) - (u-1)(\boldsymbol{\mu}_1;\boldsymbol{a})$, where $u \in \mathbb{R}$. We thus obtain that

$$\begin{aligned}(\boldsymbol{\mu}_2;\boldsymbol{b}) - (\mathbf{0}_K;\mathbf{1}_K) &= (u-1)\left((\boldsymbol{\mu}_1;\boldsymbol{a}) - (\boldsymbol{\mu}_2;\boldsymbol{b})\right)\\ (\boldsymbol{\mu}_1;\boldsymbol{a}) - (\mathbf{0}_K;\mathbf{1}_K) &= u\left((\boldsymbol{\mu}_1;\boldsymbol{a}) - (\boldsymbol{\mu}_2;\boldsymbol{b})\right).\end{aligned} \tag{B.19}$$

Furthermore, denoting the coordinates of $\boldsymbol{a'}$ and $\boldsymbol{b}$ by $\{a_i\}_{i=1}^\kappa$ and $\{b_i\}_{i=1}^K$, we similarly obtain that

$$\begin{aligned}\mathbf{0}_K &= u\boldsymbol{\mu}_2 - (u-1)\boldsymbol{\mu}_1\\ 1 &= ub_i - (u-1)a_i, \quad 1 \le i \le \kappa\\ 1 &= ub_i, \quad d+1 \le i \le K\end{aligned} \tag{B.20}$$

The last two of equations imply that

$$\sum_{i=1}^{\kappa}(a_i - b_i)^2 = \frac{\|\mathbf{1}_\kappa - \boldsymbol{a'}\|_2^2}{u^2}$$

and

$$\sum_{i=\kappa+1}^{K} b_i^2 = \frac{K-\kappa}{u^2}.$$

Combining (B.15), (B.19) and the above two equations, we rewrite the objective function of (B.18) as follows:

$$\begin{aligned}
&(\eta|u| + |u-1|(1-\eta)) \left\|(\boldsymbol{\mu}_1; \boldsymbol{a}) - (\boldsymbol{\mu}_2; \boldsymbol{b})\right\|_2 \\
=&(\eta|u| + |u-1|(1-\eta)) \sqrt{\|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\|_2^2 + \sum_{i=1}^{\kappa}(a_i - b_i)^2 + \sum_{i=\kappa+1}^{K} b_i^2} \\
\geq&(\eta|u| + |u-1|(1-\eta)) \sqrt{\epsilon^2 + \frac{\|\mathbf{1}_\kappa - \boldsymbol{a}'\|_2^2}{u^2} + \frac{K-\kappa}{u^2}} \\
=&\Bigg\{(K-\kappa)\left((1-\eta)\left|\frac{u-1}{u}\right| + \eta\right)^2 + \epsilon^2\left(\eta|u| + |u-1|(1-\eta)\right)^2 \\
&+ \left\|\mathbf{1}_\kappa - \boldsymbol{a}'\right\|_2^2 \left((1-\eta)\left|\frac{u-1}{u}\right| + \eta\right)^2\Bigg\}^{1/2},
\end{aligned} \tag{B.21}$$

where equality is achieved if and only if $\|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\|_2 = \epsilon$. One can make the following two observations: $u = 0$ does not yield a minimizer of (B.18), and for any $u \neq 0$, (B.21) obtains its minimum at $\boldsymbol{a}' = \mathbf{1}_\kappa$. In view of these observations and the derivation above, we define

$$f(u) := (K-\kappa)\left((1-\eta)\left|\frac{u-1}{u}\right| + \eta\right)^2 + \epsilon^2\left(\eta|u| + |u-1|(1-\eta)\right)^2, \tag{B.22}$$

and note that (B.18) is equivalent to

$$\min_{u\neq 0} \quad \sqrt{f(u)}. \tag{B.23}$$

We rewrite $f(u)$ as

$$f(u) = \begin{cases} (K-\kappa)\left(\frac{u-1}{u}(1-\eta) + \eta\right)^2 + \epsilon^2\left(\eta u + (1-\eta)(u-1)\right)^2, & u \geq 1 \\ (K-\kappa)\left(\frac{1-u}{u}(1-\eta) + \eta\right)^2 + \epsilon^2\left(\eta u + (1-\eta)(1-u)\right)^2, & 1 \geq u > 0 \\ (K-\kappa)\left(\frac{u-1}{u}(1-\eta) + \eta\right)^2 + \epsilon^2\left(\eta u + (1-\eta)(u-1)\right)^2, & 0 > u \end{cases}$$

We denote

$$r_1(u) := (K-\kappa)\left(\frac{u-1}{u}(1-\eta)+\eta\right)^2 + \epsilon^2\left(\eta u + (1-\eta)(u-1)\right)^2$$

and

$$r_2(u) := (K-\kappa)\left(\frac{1-u}{u}(1-\eta)+\eta\right)^2 + \epsilon^2\left(\eta u + (1-\eta)(1-u)\right)^2.$$

Their derivatives are

$$r_1'(u) = \frac{2}{u^3}\left(u-(1-\eta)\right)\left(\epsilon^2 u^3 + (K-\kappa)(1-\eta)\right)$$

and

$$r_2'(u) = \frac{2}{u^3}\left((2\eta-1)u+(1-\eta)\right)\left(\epsilon^2(2\eta-1)u^3 - (K-\kappa)(1-\eta)\right).$$

These expressions for r_1' and r_2' imply that the critical points for r_1 are

$$u_{r_1}^{(1)} = 1-\eta \quad \text{and} \quad u_{r_1}^{(2)} = -\left(\frac{(K-\kappa)(1-\eta)}{\epsilon^2}\right)^{\frac{1}{3}}$$

and the critical points for r_2 are

$$u_{r_2}^{(1)} = -\left(\frac{1-\eta}{2\eta-1}\right) \quad \text{and} \quad u_{r_2}^{(2)} = \left(\frac{(K-\kappa)(1-\eta)}{\epsilon^2(2\eta-1)}\right)^{\frac{1}{3}}.$$

We note that r_1 is increasing on $(u_{r_1}^{(2)},0)\cup(u_{r_1}^{(1)},\infty)$ and decreasing on $(-\infty,u_{r_1}^{(2)})\cup(0,u_{r_1}^{(1)})$. On the other hand, r_2 is increasing on $(u_{r_2}^{(1)},0)\cup(u_{r_2}^{(2)},\infty)$ and decreasing on $(-\infty,u_{r_2}^{(1)})\cup(0,u_{r_2}^{(2)})$. Since $\eta > \eta^\star = \frac{K-\kappa+\epsilon^2}{K-\kappa+2\epsilon^2}$, $u_{r_2}^{(2)} \in (0,1)$. The derivative of f with respect to u is

$$f_u'(u) = \begin{cases} r_1'(u), & u > 0 \\ r_2'(u), & 1 > u > 0 \\ r_1'(u), & 0 > u. \end{cases}$$

So $f(\cdot)$ is increasing on $(u_{r_1}^{(2)},0)\cup(u_{r_2}^{(2)},\infty)$ and decreasing on $(-\infty,u_{r_1}^{(2)})\cup(0,u_{r_2}^{(2)})$. The

values of f at $u_{r_2}^{(2)}$ and $u_{r_1}^{(2)}$ are

$$f(u_{r_2}^{(2)}) = \left(\left(\frac{(K-\kappa)(1-\eta)(2\eta-1)^2}{\epsilon^2} \right)^{\frac{1}{3}} + (1-\eta) \right)^2 \left((K-\kappa)^{\frac{1}{3}} \left(\frac{\epsilon^2(2\eta-1)}{(1-\eta)} \right)^{\frac{2}{3}} + \epsilon^2 \right),$$

$$f(u_{r_1}^{(2)}) = \left(\left(\frac{(K-\kappa)(1-\eta)}{\epsilon^2} \right)^{\frac{1}{3}} + (1-\eta) \right)^2 \left((K-\kappa)^{\frac{1}{3}} \left(\frac{\epsilon^2}{(1-\eta)} \right)^{\frac{2}{3}} + \epsilon^2 \right).$$

Consequently, the minimum of f is obtained at $u^\star := u_{r_2}^{(2)}$. By (B.19) and (B.20), the means $\boldsymbol{\mu}_1$, $\boldsymbol{\mu}_2$ and the covariance matrices $\boldsymbol{\Sigma}_1$, $\boldsymbol{\Sigma}_2$ satisfy: $\mathbf{0}_K = u^\star \boldsymbol{\mu_2} + (1-u^\star)\boldsymbol{\mu_1}$, $\boldsymbol{\Sigma}_1 = \mathrm{diag}(\mathbf{1}_\kappa; \mathbf{0}_{K-\kappa})$ and $\boldsymbol{\Sigma}_2 = \mathrm{diag}(\mathbf{1}_\kappa; (u^\star)^{-2}\mathbf{1}_{K-\kappa})$. Moreover, the norms of $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$ can be computed from (B.20) as $u^\star\epsilon$ and $(1-u^\star)\epsilon$, respectively.

B.3 Proof of Proposition 3.3.3

Notice that since $\boldsymbol{\Sigma}_0 \in \mathcal{S}_{++}^K$, $\det(\boldsymbol{\Sigma}_0) > 0$. On the other hand, since $\boldsymbol{\Sigma}_1 \in \mathcal{S}_{+}^K$ with $\mathrm{rank}(\boldsymbol{\Sigma}_1) = \kappa < K$, $\det(\boldsymbol{\Sigma}_1) = 0$. Therefore,

$$\log \frac{\det(\boldsymbol{\Sigma}_0)}{\det(\boldsymbol{\Sigma}_1)} = \log\det(\boldsymbol{\Sigma}_0) - \log\det(\boldsymbol{\Sigma}_1) = \infty$$

and this observation and (B.10) imply that $KL(\mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)||\mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)) = \infty$.

Appendix C

Supplementary proofs for Chapter 4

C.1 Proof of Proposition 4.3.1

First recall the property that if any two points in a given set can be connected by a continuous path lying entirely in the set, then this set must be a connected set. Now any two points $\boldsymbol{x}, \boldsymbol{y} \in \mathscr{R}^\star$ can be connected by the line segment $\{\alpha\boldsymbol{x} + (1-\alpha)\boldsymbol{y} : \alpha \in [0,1]\} \subset \mathscr{R}^\star$. Thus $\mathscr{R}^\star$ is connected.

Moreover, $Z = \mathbb{R}^{n-1} \times \{0\}$ has Lebesgue measure (denoted as μ) zero since

$$\mu(Z) = \int_{\mathbb{R}^n} \mathbb{1}_Z \, d\boldsymbol{x} = \int_{\mathbb{R}^{n-1}} \left(\int_{\{0\}} \mathbb{1}_Z \, dx_n \right) d\boldsymbol{x}_{-n} = 0. \tag{C.1}$$

Here $\mathbb{1}_Z$ is the indicator function on Z, and $\boldsymbol{x}_{-n} \in \mathbb{R}^{n-1}$ is the vector formed by the first $n-1$ coordinates of $\boldsymbol{x}$. We used Tonelli's theorem to obtain the second equality, and the fact $\int_{\{0\}} \mathbb{1}_Z \, dx_n = 0$ to obtain the third equality. The rest of (ii) is straightforward.

For (iii), suppose that there is another point $\widetilde{\boldsymbol{x}} \in \mathscr{R}^\star \setminus \{\boldsymbol{x}\}$ which can represent $\boldsymbol{x}$ up to a global sign flipping. Since both $\boldsymbol{x}$ and $\widetilde{\boldsymbol{x}}$ are in $\mathscr{R}^\star$, which means they need to have the same sign for the last component, it must be $\boldsymbol{x} = \widetilde{\boldsymbol{x}}$. We get a contradiction.

C.2 Proof of Proposition 4.3.2

We prove by contradiction. Suppose that there is a $\boldsymbol{x}' \in \mathbb{C}^{n-1}$ but $\boldsymbol{x}' \notin B$. Then for any $x_1 \in \mathbb{R}_+$, $\boldsymbol{x} = (x_1; \boldsymbol{x}') \notin \mathbb{R}_+ \times B = \mathscr{R}$ and $\boldsymbol{x} \in \mathbb{C}^{n-1} \setminus Z$. Since $\mathscr{R}$ is representative, we can find a $\theta \in [0, 2\pi)$ and $\widetilde{\boldsymbol{x}} \in \mathscr{R}$ so that

$$\mathrm{e}^{i\theta}\boldsymbol{x} = \widetilde{\boldsymbol{x}}. \tag{C.2}$$

Since $\mathscr{R}$ has the first coordinate to be positive real numbers, by looking at the first component of equation (C.2) we have

$$\begin{cases} x_1 \cos\theta > 0 \\ x_1 \sin\theta = 0 \end{cases}, \tag{C.3}$$

from where we deduce that $\theta = 0$ and so $\boldsymbol{x} = \widetilde{\boldsymbol{x}} \in \mathscr{R}$. This contradicts our construction that $\boldsymbol{x} \notin \mathscr{R}$.

C.3 Proof of Proposition 4.3.3

First, Z has measure zero due to the same reason as in (C.1). Next, it is clear that any two points $\boldsymbol{x}, \boldsymbol{y} \in \mathscr{R}^\star$ can be connected by the line segment $\{\alpha\boldsymbol{x} + (1-\alpha)\boldsymbol{y} : \alpha \in [0,1]\} \subset \mathscr{R}^\star$, and so $\mathscr{R}^\star$ is a connected set. To see $\mathscr{R}^\star$ is representative, for any $\boldsymbol{x} = (r_1 e^{i\theta_1}, x_2, \ldots, x_n) \in \mathbb{C}^n \setminus Z$ where $r_1 > 0$, one can choose $\theta = 2\pi - \theta_1$ so that $e^{i\theta}\boldsymbol{x} \in \mathscr{R}^\star$. To show it is also smallest, we use a similar argument to that in (4.3.2). Let $\boldsymbol{x} \in \mathscr{R}^\star$ where we write $\boldsymbol{x} = (x_1; \boldsymbol{x}')$ with $\boldsymbol{x}' \in \mathbb{C}^{n-1}$. If another element $\widetilde{\boldsymbol{x}} \neq \boldsymbol{x} \in \mathscr{R}^\star$ can be represented by $\boldsymbol{x}$, namely, if there is $\theta \in [0, 2\pi)$ such that $\widetilde{\boldsymbol{x}} = \mathrm{e}^{i\theta}\boldsymbol{x}$, then we need to have $\mathrm{Im}(\mathrm{e}^{i\theta}x_1) = 0$ and $\mathrm{Re}(\mathrm{e}^{i\theta}x_1) > 0$. That is,

$$\begin{cases} x_1 \cos\theta > 0 \\ x_1 \sin\theta = 0 \end{cases}. \tag{C.4}$$

Since $x_1 > 0$, (C.4) implies that $\theta = 0$. But this contradicts with that $\boldsymbol{x} \neq \widetilde{\boldsymbol{x}}$ and thus no element in $\mathscr{R}^\star$ can be represented by a distinct element in $\mathscr{R}^\star$.

C.4 Proof of Proposition 4.4.1

It is clear that $\mathcal{R}$ is connected in $\mathbb{S}^{m_1 m_2}$ since $\mathscr{R}$ is path-connected set on $\mathbb{S}^{m_1 \times m_2}$ with the inherited subspace Euclidean topology of $\mathbb{C}^{m_1 \times m_2}$. Also, $\mathcal{N}$ is of Lebesgue measure 0 since it is a product of finite points. Now we are going to prove $\mathcal{R}$ is a representative of $\mathbb{S}^{m_1 m_2}$.

Let $\mathcal{G}$ be the set of all possible symmetry transfers composed of sequences of global phase transfer and global phase conjugation in PR. For any given $\boldsymbol{z} = (\mathrm{e}^{i\theta_0}, \mathrm{e}^{i\theta_1}, \cdots, \mathrm{e}^{i\theta_{m_1 m_2 - 1}})$, we need to find a $\boldsymbol{\omega} \in \mathcal{R}$ such that there is a $g \in \mathcal{G}$ satisfying $g * \boldsymbol{\omega} = \boldsymbol{z}$. If $\mathrm{Im}(\mathrm{e}^{i(\theta_1 - \theta_0)}) > 0$, we take $\boldsymbol{\omega} = (1, \mathrm{e}^{i(\theta_1 - \theta_0)}, \mathrm{e}^{i(\theta_2 - \theta_0)}, \cdots, \mathrm{e}^{i(\theta_{m_1 m_2 - 1} - \theta_0)})$ then $\boldsymbol{\omega} \in \mathcal{R}$ and $\mathrm{e}^{i\theta_0}\boldsymbol{\omega} = \boldsymbol{z}$. On the other hand, if $\mathrm{Im}(\mathrm{e}^{i(\theta_1 - \theta_0)}) < 0$, we can consider the conjugate format $\boldsymbol{\omega} = (1, \overline{\mathrm{e}^{i(\theta_1 - \theta_0)}}, \overline{\mathrm{e}^{i(\theta_2 - \theta_0)}}, \cdots, \overline{\mathrm{e}^{i(\theta_{m_1 m_2 - 1} - \theta_0)}}) \in \mathcal{R}$ and obviously a global phase negation followed by a global phase transfer $e^{i\theta_0}$ leads to $\boldsymbol{z}$. This proves that $\mathcal{R}$ is representative.

At last, we need to show the smallestness in the sense that with any point of $\mathcal{R}$ removed, we cannot recover it by other points in $\mathcal{R}$. That is, with arbitrary $\tilde{\boldsymbol{z}} \in \mathcal{R}$ given, for all $g \in \mathcal{G}$ and all $\boldsymbol{z} \in \mathcal{R} \setminus \{\tilde{\boldsymbol{z}}\}$, we have $g * \boldsymbol{z} \neq \tilde{\boldsymbol{z}}$.

We first claim that any g is equivalent to an optional global phase conjugation followed by a global phase transfer. To see this, it is sufficient to prove that the order of phase conjugation and phase transfer can be exchanged. Let ψ denote a global phase transfer by $e^{i\psi}$ and f phase conjugation. Now if $\psi \circ f = f \circ \psi'$, or $-(\psi' + \theta) = -\theta + \psi + 2k\pi$, we have $\psi' = -\psi - 2k\pi$. So one can keep exchanging conjugation and transfer so that all conjugations precede transfers. The conjugations now can be equivalently written as an optional conjugation, and the transfers as a single transfer.

Now we can go back to the proof of smallestness. Write $\tilde{\boldsymbol{z}} = (\mathrm{e}^{i\tilde{\theta_0}}, \mathrm{e}^{i\tilde{\theta_1}}, \cdots, \mathrm{e}^{i\tilde{\theta}_{m_1 m_2 - 1}})$ and $\boldsymbol{z} = (\mathrm{e}^{i\theta_0}, \mathrm{e}^{i\theta_1}, \cdots, \mathrm{e}^{i\theta_{m_1 m_2 - 1}})$ where $\tilde{\theta_0} = \theta_0 = 0$ and $\mathrm{Im}(\mathrm{e}^{i\tilde{\theta_1}})$, $\mathrm{Im}(\mathrm{e}^{i\theta_1}) > 0$. Suppose that there is a $g \in \mathcal{G}$ such that $\tilde{\boldsymbol{z}} = g * \boldsymbol{z}$. we may assume $g = f \circ \psi$ or $g = \psi$ where ψ is a phase transition with the total angles ψ and f is the conjugate flipping. If $g = f \circ \psi$, $\tilde{\boldsymbol{z}} = g * \boldsymbol{z}$ implies that

$$\tilde{\theta}_j \equiv -(\psi + \theta_j) + 2\pi k_j \mod 2\pi \quad \forall j \tag{C.5}$$

for some $k_j \in \mathbb{Z}$. We can solve $\psi = 2\pi k_0$ as $j = 0$ and this implies $\tilde{\theta}_j \equiv 2\pi(k_j - k_0) - \theta_j$

mod $2\pi \equiv -\theta_j$ for all j, especially, $\tilde{\theta}_1 = -\theta_1$. This contradict with the fact that $\mathrm{Im}(\mathrm{e}^{i\tilde{\theta}_1})$, $\mathrm{Im}(\mathrm{e}^{i\theta_1}) > 0$. If $g = \psi$, we then have the relationship

$$\tilde{\theta}_j \equiv (\psi + \theta_j) + 2\pi k_j \mod 2\pi. \tag{C.6}$$

Again, we can solve $\psi = -2\pi k_0$ as $j = 0$ and this indicates that $\tilde{\boldsymbol{z}} = \boldsymbol{z}$ which contradicts the assumption. Hence, we prove the smallestness.

Appendix D

Brief description of measurements

D.1 Description of metrics for anomaly detection tasks

AUC (area-under-curve) is the area under the Receiver Operating Characteristic (ROC) curve. Recall that the True Positive Rate (TPR), or Recall, is the number of samples correctly labeled as positive divided by the total number of actual positive samples. The False Positive Rate (FPR), on the other hand, is the number of negative samples incorrectly labeled as positive divided by the total number of actual negative samples. The ROC curve is a graph of TPR as a function of FPR.

AP (average-precision) is the area under the Precision-Recall Curve. While Recall is the TPR, Precision is the number of samples correctly labeled as positive divided by the total number of predicted positives. The Precision-Recall curve is the graph of Precision as a function of Recall.

Both AUC and AP can be computed using the corresponding functions in the scikit-learn package [143].

D.2 Description of the mean Square Error (MSE) measurement for FPR

Our reconstructed image is in $\mathbb{C}^{m\times m}$, where our original image is in $\mathbb{C}^{n\times n}$. To account for the three symmetries when taking MSE measure, we take the following steps: we take the original image, and scan through the larger reconstructed image to account for the translation symmetry. At each scan position, we calculate an adjusted MSE between the current patch $\boldsymbol{B} \in \mathbb{C}^{n\times n}$ and the original image $\boldsymbol{A}$. A $\lambda > 0$ and a global phase factor $\mathrm{e}^{i\theta}$ (to account for the global phase) are introduced when calculating the MSE, i.e.,

$$\min_{\theta,\eta\geq 0} \left\| \boldsymbol{A} - \eta \boldsymbol{B} \mathrm{e}^{i\theta} \right\|_F^2. \tag{D.1}$$

The smallest adjusted MSE is recorded over all scan positions. Then, the original image $\boldsymbol{A}$ is 2D flipped and the same scanning process is repeated to calculate another smallest MSE, to account for the flipping symmetry. The smaller of the smallest MSE values is finally taken.

Below, we show that the optimal value in (D.1) can be easily computed. First we expand the square inside the objective and perform partial minimization with respect to θ, leading to

$$\max_{\theta} \mathrm{Re} \left\langle \boldsymbol{A}, \boldsymbol{B} \mathrm{e}^{i\theta} \right\rangle. \tag{D.2}$$

But $\mathrm{Re}\left\langle \boldsymbol{A}, \boldsymbol{B}\mathrm{e}^{i\theta}\right\rangle = \mathrm{Re}\left(\langle \boldsymbol{A}, \boldsymbol{B}\rangle \, \mathrm{e}^{i\theta}\right) \leq \left|\langle \boldsymbol{A}, \boldsymbol{B}\rangle \, \mathrm{e}^{i\theta}\right| \leq |\langle \boldsymbol{A}, \boldsymbol{B}\rangle|$ and the upper bound is achievable when $\theta = -\angle \langle \boldsymbol{A}, \boldsymbol{B}\rangle$. So the optimization problem now becomes

$$\min_{\eta\geq 0} \ \|\boldsymbol{A}\|_F^2 + \eta^2 \|\boldsymbol{B}\|_F^2 - 2\eta |\langle \boldsymbol{A}, \boldsymbol{B}\rangle|. \tag{D.3}$$

The minimum of equation (D.3) occurs either when $\eta = 0$, which is $\|\boldsymbol{A}\|_F^2$, or when $2\eta\|\boldsymbol{B}\|_F^2 = 2|\langle \boldsymbol{A}, \boldsymbol{B}\rangle| \Longrightarrow \eta = |\langle \boldsymbol{A}, \boldsymbol{B}\rangle|/\|\boldsymbol{B}\|_F^2$, leading to the function value

$$\|\boldsymbol{A}\|_F^2 - \frac{|\langle \boldsymbol{A}, \boldsymbol{B}\rangle|}{\|\boldsymbol{B}\|_F^2}, \tag{D.4}$$

which is the smaller one.

Appendix E

Numerical results of experiments for Chapter 3

We present as tables the numerical values depicted in Figures 3.2 and 3.3 in §E.1 and those in Figure 3.5 in §E.2.

E.1 Table representation for Figures 3.2 and 3.3

Tables E.1-E.12 report the averaged AUC and AP scores with training ratio of outliers per inliers $c \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$ that were depicted in Figures 3.2 and 3.3. Each table describes one of the averaged scores (AUC or AP) for one of the six datasets (COVID-19, CIFAR-10, Caltech101, Fashion MNIST, KDDCUP-99 and Reuters-21578) and also indicates the standard deviation of each value. The outperforming methods are marked in bold.

Table E.1: AUC scores of COVID-19.

	Training ratio of outliers per inliers, c				
Methods	0.1	0.2	0.3	0.4	0.5
MAW	**0.682** ± 0.021	**0.639** ± 0.018	**0.606** ± 0.020	0.551 ± 0.030	0.534 ± 0.010
DAGMM	0.547± 0.068	0.565± 0.051	0.538 ± 0.062	0.524 ± 0.060	0.523 ± 0.057
DSEBMs	0.471± 0.000	0.471± 0.000	0.471 ± 0.000	0.471 ± 0.000	0.471 ± 0.000
IF	0.604	0.571	0.555	0.523	0.499
LOF	0.672	0.618	0.572	**0.580**	**0.589**
OCGAN	0.492± 0.000	0.492± 0.000	0.492 ± 0.000	0.485 ± 0.000	0.491 ± 0.000
OCSVM	0.528	0.528	0.528	0.535	0.521
RSRAE	0.565± 0.031	0.527± 0.028	0.476 ± 0.023	0.454 ± 0.018	0.427 ± 0.011

Table E.2: AP scores of COVID-19.

	Training ratio of outliers per inliers, c				
Methods	0.1	0.2	0.3	0.4	0.5
MAW	0.459 ± 0.014	**0.442** ± 0.011	**0.424** ± 0.018	0.368 ± 0.015	0.353 ± 0.013
DAGMM	0.354± 0.053	0.390 ± 0.057	0.316 ± 0.052	0.357 ± 0.050	0.348 ± 0.047
DSEBMs	0.372± 0.000	0.375 ± 0.000	0.364 ± 0.000	0.360 ± 0.000	0.358 ± 0.000
IF	0.425	0.404	0.392	0.373	0.363
LOF	**0.463**	0.422	0.402	**0.374**	**0.371**
OCGAN	0.381± 0.000	0.381 ± 0.000	0.381 ± 0.000	0.373 ± 0.000	0.350 ± 0.000
OCSVM	0.315	0.315	0.315	0.372	0.365
RSRAE	0.388± 0.018	0.377 ± 0.016	0.355 ± 0.011	0.352 ± 0.010	0.340 ± 0.009

Table E.3: AUC scores of CIFAR-10.

	Training ratio of outliers per inliers, c				
Methods	0.1	0.2	0.3	0.4	0.5
MAW	0.621 ± 0.013	**0.609** ± 0.014	**0.607** ± 0.012	0.600 ± 0.010	**0.595** ± 0.013
LOF	0.582	0.574	0.559	0.551	0.539
OCSVM	0.595	0.587	0.580	0.564	0.570
IF	0.603	0.586	0.596	0.581	0.569
RSRAE	**0.638** ± 0.010	0.607 ± 0.017	0.599 ± 0.023	**0.610** ± 0.025	0.589 ± 0.023
DSEBMs	0.586 ± 0.006	0.584 ±0.006	0.580 ± 0.004	0.576± 0.006	0.556 ±0.006
OCGAN	0.501 ±0	0.501 ±0	0.499 ±0	0.487 ± 0	0.476 ±0
DAGMM	0.574 ± 0.030	0.557 ± 0.035	0.541 ±0.037	0.510 ± 0.0331	0.545 ± 0.037

Table E.4: AP scores of CIFAR-10.

	Training ratio of outliers per inliers, c				
Methods	0.1	0.2	0.3	0.4	0.5
MAW	0.427 ± 0.010	**0.419** ±0.012	0.414 ± 0.011	**0.400**±0.009	**0.411** ±0.011
LOF	0.395	0.036	0.377	0.374	0.371
OCSVM	0.408	0.400	0.393	0.378	0.385
IF	0.416	0.395	0.403	0.389	0.373
RSRAE	**0.434** ± 0.011	0.412 ±0.020	**0.417** ± 0.022	0.391 ± 0.019	0.400 ± 0.014
DSEBMs	0.391 ±0.008	0.388 ± 0.008	0.386 ±0.004	0.382 ± 0.006	0.379±0.003
OCGAN	0.342 ± 0	0.340±0	0.339 ±0	0.337 ± 0	0.335±0
DAGMM	0.378 ±0.049	0.369 ±0.041	0.355 ± 0.030	0.308 ± 0.026	0.352 ± 0.047

Table E.5: AUC scores of Caltech101.

	Training ratio of outliers per inliers, c				
Methods	0.1	0.2	0.3	0.4	0.5
MAW	**0.801** ± 0.017	**0.760** ± 0.028	**0.700** ± 0.038	**0.608** ± 0.031	**0.570** ± 0.021
DAGMM	0.684 ± 0.100	0.588 ± 0.115	0.500± 0.100	0.509 ± 0.101	0.514 ± 0.095
DSEBMs	0.536 ± 0.011	0.612± 0.025	0.577 ± 0.030	0.564 ± 0.021	0.536 ± 0.021
IF	0.755	0.694	0.626	0.575	0.540
LOF	0.674	0.593	0.495	0.436	0.411
OCGAN	0.494 ± 0.000	0.494± 0.000	0.494± 0.000	0.500 ± 0.000	0.500 ± 0.000
OCSVM	0.682	0.618	0.577	0.538	0.516
RSRAE	0.774 ± 0.027	0.722 ± 0.041	0.664 ± 0.082	0.579 ± 0.047	0.568 ± 0.036

Table E.6: AP scores of Caltech101.

	Training ratio of outliers per inliers, c				
Methods	0.1	0.2	0.3	0.4	0.5
MAW	**0.634** ± 0.027	**0.572** ± 0.039	**0.531** ± 0.064	0.412 ± 0.029	0.414 ± 0.021
DAGMM	0.574± 0.088	0.422 ± 0.112	0.308 ± 0.102	0.351 ± 0.074	0.363 ± 0.076
DSEBMs	0.385± 0.003	0.472± 0.051	0.398±0.019	0.383 ± 0.023	0.365 ± 0.028
IF	0.545	0.486	0.430	0.304	0.371
LOF	0.460	0.400	0.337	0.304	0.290
OCGAN	0.362± 0.000	0.362± 0.000	0.362 ± 0.000	0.362 ± 0.000	0.362 ± 0.000
OCSVM	0.472	0.419	0.380	0.352	0.339
RSRAE	0.595± 0.038	0.551 ± 0.045	0.495 ±0.073	**0.425** ± 0.040	**0.443** ± 0.027

Table E.7: AUC scores of Fashion MNIST

	Training ratio of outliers per inliers, c				
Methods	0.1	0.2	0.3	0.4	0.5
MAW	**0.897** ± 0.013	**0.879** ± 0.011	**0.852** ± 0.022	0.830 ± 0.017	0.801 ± 0.016
DAGMM	0.607 ± 0.093	0.376 ± 0.070	0.427 ± 0.090	0.401 ± 0.078	0.411 ± 0.081
DSEBMs	0.730 ± 0.092	0.729 ± 0.105	0.739 ± 0.086	0.723 ± 0.106	0.687 ± 0.096
IF	0.893	0.875	0.843	**0.834**	**0.827**
LOF	0.569	0.507	0.476	0.468	0.458
OCGAN	0.542 ± 0.006	0.538 ± 0.004	0.544 ± 0.014	0.531 ± 0.003	0.525 ± 0.004
OCSVM	0.895	0.874	0.848	0.831	0.814
RSRAE	0.860 ± 0.022	0.848 ± 0.022	0.829 ± 0.042	0.831 ± 0.028	0.808 ± 0.028

Table E.8: AP scores of Fashion MNIST

	Training ratio of outliers per inliers, c				
Methods	0.1	0.2	0.3	0.4	0.5
MAW	**0.788** ±0.013	0.754 ± 0.014	0.723±0.029	0.686 ± 0.025	0.672 ±0.021
DAGMM	0.482 ±0.051	0.303 ±0.057	0.334 ±0.113	0.318 ±0.056	0.330 ± 0.038
DSEBMs	0.600 ± 0.045	0.609± 0.120	0.613±0.089	0.605 ±0.086	0.565 ± 0.072
IF	0.768	0.724	0.693	0.665	0.642
LOF	0.382	0.331	0.308	0.301	0.294
OCGAN	0.504 ± 0.002	0.503 ± 0.003	0.500 ± 0.059	0.495 ± 0.001	0.493 ± 0.001
OCSVM	0.801	**0.768**	**0.735**	**0.696**	0.664
RSRAE	0.749 ± 0.029	0.736 ± 0.032	0.716 ± 0.048	0.683 ± 0.036	**0.680** ± 0.042

Table E.9: AUC scores of KDDCUP-99.

	Training ratio of outliers per inliers, c				
Methods	0.1	0.2	0.3	0.4	0.5
MAW	**0.765**± 0.025	**0.732** ± 0.015	**0.647** ± 0.012	**0.594** ± 0.014	0.556 ± 0.014
DAGMM	0.446 ± 0.047	0.506 ± 0.064	0.459 ± 0.087	0.373 ± 0.109	0.464 ± 0.998
DSEBMs	0.450 ± 0.000	0.447 ± 0.000	0.446 ± 0.000	0.444 ± 0.000	0.444 ± 0.000
IF	0.636	0.6331	0.562	0.493	0.457
LOF	0.391	0.407	0.392	0.394	0.391
OCGAN	0.582 ± 0.132	0.472 ± 0.163	0.525 ± 0.133	0.418 ± 0.136	0.535 ± 0.133
OCSVM	0.543	0.598	0.595	0.438	0.426
RSRAE	0.704 ± 0.048	0.698 ± 0.050	0.606 ± 0.065	0.584 ± 0.034	**0.574** ± 0.046

Table E.10: AP scores of KDDCUP-99.

	Training ratio of outliers per inliers, c				
Methods	0.1	0.2	0.3	0.4	0.5
MAW	**0.765**± 0.025	**0.732** ± 0.015	**0.647** ± 0.012	**0.594** ± 0.014	0.556 ± 0.014
DAGMM	0.446 ± 0.047	0.506 ± 0.064	0.459 ± 0.087	0.373 ± 0.109	0.464 ± 0.998
DSEBMs	0.450 ± 0.000	0.447 ± 0.000	0.446 ± 0.000	0.444 ± 0.000	0.444 ± 0.000
IF	0.636	0.6331	0.562	0.493	0.457
LOF	0.391	0.407	0.392	0.394	0.391
OCGAN	0.582 ± 0.132	0.472 ± 0.163	0.525 ± 0.133	0.418 ± 0.136	0.535 ± 0.133
OCSVM	0.543	0.598	0.595	0.438	0.426
RSRAE	0.704 ± 0.048	0.698 ± 0.050	0.606 ± 0.065	0.584 ± 0.034	**0.574** ± 0.046

Table E.11: AUC scores of Reuters-21578.

	Training ratio of outliers per inliers, c				
Methods	0.1	0.2	0.3	0.4	0.5
MAW	0.885 ± 0.028	**0.830** ± 0.013	0.770 ± 0.017	**0.700** ± 0.002	**0.648** ± 0.016
DAGMM	0.500 ± 0.000	0.511 ± 0.027	0.566 ± 0.110	0.559 ± 0.087	0.570 ± 0.091
DSEBMs	**0.887** ± 0.012	0.825 ± 0.012	**0.790** ± 0.015	0.690 ± 0.002	0.648 ± 0.010
IF	0.544	0.535	0.520	0.453	0.452
LOF	0.757	0.612	0.579	0.631	0.616
OCGAN	0.648 ± 0.127	0.477 ± 0.129	0.498 ± 0.140	0.519 ± 0.132	0.502 ± 0.099
OCSVM	0.882	0.817	0.785	0.673	0.640
RSRAE	0.786 ± 0.042	0.755 ± 0.034	0.716 ± 0.033	0.605 ± 0.001	0.494 ± 0.004

Table E.12: AP scores of Reuters-21578.

	Training ratio of outliers per inliers, c				
Methods	0.1	0.2	0.3	0.4	0.5
MAW	0.755 ± 0.041	0.677 ± 0.026	0.627 ± 0.029	**0.518** ± 0.004	0.474 ± 0.013
DAGMM	0.316 ± 0.000	0.316 ± 0.013	0.365 ± 0.020	0.362 ± 0.015	0.372 ± 0.012
DSEBMs	**0.763** ± 0.012	**0.697** ± 0.011	**0.666** ± 0.007	0.515 ± 0.003	0.473 ± 0.003
IF	0.368	0.372	0.365	0.301	0.298
LOF	0.580	0.438	0.421	0.498	**0.486**
OCGAN	0.408 ± 0.045	0.334 ± 0.098	0.365 ± 0.106	0.504 ± 0.083	0.497 ± 0.094
OCSVM	0.746	0.681	0.637	0.467	0.438
RSRAE	0.593 ± 0.051	0.563 ± 0.035	0.488 ± 0.036	0.403± 0.001	0.415 ± 0.003

E.2 Table representation for Figure 3.5

Tables E.13-E.16 record the averaged AUC and AP scores with training ratio of outliers per inliers c = 0.1, 0.2, 0.3, 0.4 and 0.5 that were depicted in Figure 3.5. Each table describes one of the averaged scores (AUC or AP) for one of the two representative datasets (KDDCUP-99 and COVID-19) and also indicates the standard deviation of each value. The outperforming methods are marked in bold.

Table E.13: AUC scores of KDD-99 for variations of MAW

	Training ratio of outliers per inliers, c				
Methods	0.1	0.2	0.3	0.4	0.5
MAW	**0.945** ± 0.028	**0.906** ± 0.018	**0.832** ± 0.016	**0.775** ± 0.023	**0.731** ± 0.017
MAW-MSE	0.844 ± 0.039	0.812 ± 0.032	0.746 ± 0.044	0.709 ± 0.020	0.675 ± 0.014
MAW-KL divergence	0.905 ± 0.026	0.863 ± 0.028	0.801 ± 0.029	0.752 ± 0.016	0.696 ± 0.018
MAW-same rank	0.912 ± 0.023	0.868 ± 0.011	0.797 ± 0.022	0.750 ± 0.012	0.699 ± 0.040
MAW-single Gaussian	0.914 ± 0.016	0.862 ± 0.021	0.796 ± 0.013	0.751 ± 0.040	0.701 ± 0.045
MAW-diagonal cov.	0.918 ± 0.023	0.858 ± 0.020	0.801 ± 0.044	0.743 ± 0.017	0.703 ± 0.015
VAE	0.821 ± 0.048	0.785 ± 0.027	0.732 ± 0.046	0.717 ± 0.018	0.685 ± 0.027

Table E.14: AP scores of KDDCUP-99 for variations of MAW

	Training ratio of outliers per inliers, c				
Methods	0.1	0.2	0.3	0.4	0.5
MAW	**0.765** ± 0.025	**0.732** ± 0.015	**0.647** ± 0.012	**0.594**± 0.014	**0.556** ± 0.014
MAW-MSE	0.715 ± 0.079	0.589 ± 0.058	0.524 ± 0.053	0.463±0.042	0.410 ± 0.028
MAW-KL divergence	0.735 ± 0.028	0.676 ± 0.028	0.618 ± 0.024	0.579±0.023	0.509±0.017
MAW-same rank	0.725 ± 0.028	0.681 ±0.015	0.622 ± 0.024	0.572 ±0.017	0.532 ±0.038
MAW-single Gaussian	0.737 ± 0.018	0.675 ±0.023	0.620 ± 0.025	0.569±0.036	0.519 ±0.044
MAW-diagonal cov.	0.724 ± 0.021	0.678 ± 0.035	0.589 ± 0.064	0.546 ±0.019	0.512 ±0.016
VAE	0.642 ± 0.030	0.555 ± 0.043	0.524 ± 0.028	0.478±0.024	0.450±0.015

Table E.15: AUC scores of COVID-19 for variations of MAW

	Training ratio of outliers per inliers, c				
Methods	0.1	0.2	0.3	0.4	0.5
MAW	**0.652** ± 0.021	**0.609** ± 0.018	**0.576** ± 0.019	**0.531** ±0.020	**0.504**±0.010
MAW-MSE	0.602 ± 0.022	0.554 ±0.063	0.528 ± 0.041	0.507± 0.014	0.479±0.021
MAW-KL divergence	0.614 ± 0.025	0.580 ± 0.026	0.508 ± 0.064	0.476 ± 0.023	0.463±0.016
MAW-same rank	0.604 ± 0.031	0.574 ± 0.048	0.527 ± 0.044	0.430 ± 0.017	0.408±0.021
MAW-single Gaussian	0.621 ± 0.027	0.586±0.029	0.507 ± 0.047	0.492±0.021	0.472±0.019
MAW-diagonal cov.	0.600 ± 0.029	0.586± 0.030	0.535 ± 0.035	0.446 ±0.028	0.439 ±0.038
VAE	0.619 ± 0.073	0.565±0.065	0.522 ± 0.049	0.508 ± 0.023	0.473±0.016

Table E.16: AP scores of COVID-19 for variations of MAW

	Training ratio of outliers per inliers, c				
Methods	0.1	0.2	0.3	0.4	0.5
MAW	**0.459** ± 0.014	**0.442** ± 0.011	**0.424** ± 0.018	**0.368**±0.015	**0.353**±0.013
MAW-MSE	0.421 ± 0.015	0.395 ± 0.025	0.377 ± 0.012	0.332±0.013	0.328 ± 0.020
MAW-KL divergence	0.427 ± 0.016	0.403± 0.012	0.370 ± 0.021	0.322±0.017	0.313 ±0.013
MAW-same rank	0.422 ± 0.021	0.413 ±0.026	0.375 ± 0.019	0.344 ±0.023	0.335 ±0.017
MAW-single Gaussian	0.425± 0.019	0.409 ± 0.012	0.374 ± 0.016	0.339± 0.014	0.329±0.016
MAW-diagonal cov.	0.412 ± 0.016	0.397 ± 0.018	0.369 ± 0.012	0.343±0.009	0.330 ±0.009
VAE	0.412 ± 0.030	0.411 ± 0.043	0.379 ± 0.028	0.341±0.011	0.333±0.013